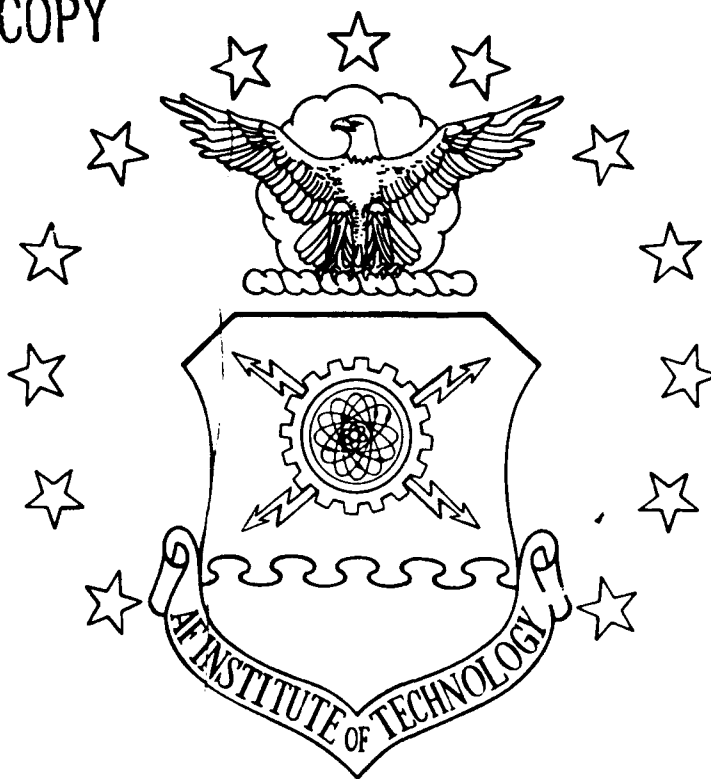


DTIC FILE COPY

①

AD-A230 598



DTIC
ELECTE
JAN 07 1991

S
D
D

A LINEAR QUADRATIC REGULATOR
WEIGHT SELECTION ALGORITHM
FOR ROBUST POLE ASSIGNMENT

THESIS

Jeffrey D. Robinson, 1Lt, USAF

AFIT/GAE/ENY/90D-23

DISTRIBUTION STATEMENT A

Approved for public release
Distribution Unlimited

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GAE/ENY/90D-23

61

A LINEAR QUADRATIC REGULATOR
WEIGHT SELECTION ALGORITHM
FOR ROBUST POLE ASSIGNMENT

THESIS

Jeffrey D. Robinson, 1Lt, USAF

AFIT/GAE/ENY/90D-23

Approved for public release; distribution unlimited

A LINEAR QUADRATIC REGULATOR WEIGHT SELECTION
ALGORITHM FOR ROBUST POLE ASSIGNMENT

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the
Requirements for the Degree of
Master of Science

by

Jeffrey D. Robinson, B.S.A.E.

First Lieutenant, USAF

December 1990

Accession For	
NTIS CR221	↓
DTIC F42	↓
Unclassified	↓
Justification	
By	
Date of issue	
Approved by	
Dist	Availability
A-1	Unlimited

Acknowledgements

This thesis could never have been finished without the tireless support and encouragement of my wife Cheralynne. I would like to express my deep appreciation for her long hours of editing and typing, and for sacrificing her free time to manage the household so I could write this thesis. I would also like to thank Bart Graham for his help with the computer software. Finally, I would like to thank Dr Brad Liebst for his patience and many helpful suggestions that made this thesis possible.

Table of Contents

List of Figures	v
List of Tables	vii
List of Symbols	viii
Abstract	x
I. Introduction	1
Background	2
Problem Statement	5
Organization	8
II. Theory	9
Controllability and Observability	9
Linear Quadratic Regulator	10
LQR Stability Margins	12
SISO Margins	12
MIMO Margins	14
III. Development of a Robust Pole Placement Algorithm	18
Defining the Problem	18
Algorithm	20
Using the Algorithm With MATLAB	20
IV. Examples	23
Example 1 - First Order SISO Case	23
Problem Setup	23
Stable Case	27
Unstable Case	27
Algorithm Results	28
Example 2 - Second Order SISO Case	33
Problem Setup	33
Finding the LQR Regions	36
Algorithm Results	41
Example 3 - Third Order SISO Case	44

Problem Setup	44
Case 1	45
Case 2	46
Case 3	51
Example 4 - Comparison with Harvey and Stein method	54
Example 5 - Aircraft Flying Qualities Improvement	65
V. Conclusions and Recommendations	70
Appendix A. MATLAB m-files	72
Appendix B. Data for the A-4D Aircraft	77
Bibliography	79
Vita	81

List of Figures

Figure 1 State Feedback Diagram	3
Figure 2 LQR SISO Block Diagram	13
Figure 3 Polar Plot of LQR Margins	15
Figure 4 Root Locus for LQR Achievable Poles	26
Figure 5 Bode Plot for Stable Case	28
Figure 6 Bode Plot for Unstable Case $k=12$	29
Figure 7 Bode Plot for Unstable Case $k=2a$	29
Figure 8 Algorithm Plot for Stable Case	31
Figure 9 Algorithm Plot for Unstable Case with Desired Pole at -7	31
Figure 10 Algorithm Plot for Unstable Case with Desired Pole at -4	32
Figure 11 Second Order SISO Example	33
Figure 12 Block Diagram for Second Order SISO	35
Figure 13 Relationship of Pole to Parameters	38
Figure 14 LQR Barriers	38
Figure 15 Bode Plot for Second Order SISO System	40
Figure 16 Nyquist Plot for LQR Design - Second Order System	42
Figure 17 Comparison of LQR to 60° PM plot	42
Figure 18 Algorithm Plot With LQR Barriers	43
Figure 19 Third Order System Block Diagram	44
Figure 20 Bode Plot for Desired Poles - Case 1	47

Figure 21	Algorithm Plot - Case 1	47
Figure 22	Bode Plot for Desired Poles - Case 2	48
Figure 23	Algorithm Plot - Case 2 (no Weighting on Actuator)	48
Figure 24	Algorithm Plot - Case 2 (Actuator Weighted)	50
Figure 25	Bode Plot for Achievable Poles - Case 2 (Actuator Weighted)	50
Figure 26	Bode Plot for Desired Poles - Case 3	52
Figure 27	Algorithm Plot - Case 3	52
Figure 28	Bode Plot for Achievable Poles - Case 3	53
Figure 29	Comparison of Desired and Achievable Poles	54
Figure 30	Comparison of Desired and Achievable Poles	58
Figure 31	Desired and Actuator Poles - Expanded View	58
Figure 32	Singular Value Plots for Example 4	59
Figure 33	Minimum Singular Value Test 1 for Harvey and Stein Example	60
Figure 34	Minimum Singular Value Test 1 for Algorithm	60
Figure 35	Minimum Singular Value Test 2 for Harvey and Stein	61
Figure 36	Minimum Singular Value Test 2 for Algorithm	61
Figure 37	Singular Value Test for a Diagonal \mathbf{R}	62
Figure 38	Singular Value Test for a Symmetric \mathbf{R}	62
Figure 39	Short Period Frequency Requirements - Category B [23:541]	67
Figure 40	Bode Plot for Desired Poles for Example 5 Aircraft	69
Figure 41	Bode Plot for Achievable Poles for Example 5 Aircraft	69

List of Tables

Table I First Order SISO Results	32
Table II Second Order SISO Results	41
Table III Third Order System Case 2 Results	51
Table IV Third Order System Case 3 Results	53
Table V Example Four Results	57

List of Symbols

A	-	state matrix
A*	-	closed loop state matrix
B	-	input matrix
C	-	output matrix
GH()	-	open loop transfer function in terms of ()
H	-	square root of state weighting matrix Q
I	-	identity matrix
jω	-	frequency domain parameter
J	-	Linear Quadratic Regulator performance index
J'	-	eigenvalue difference cost function
K	-	feedback gain matrix
K_{opt}	-	optimal feedback gain matrix from LQR design
m	-	mass
M_c	-	controllability matrix
M_o	-	observability matrix
P	-	Riccati equation solution matrix
q	-	pitch rate
Q	-	state weighting matrix
R	-	control weighting matrix
s	-	Laplace domain parameter

t	-	time
u	-	velocity
\mathbf{u}	-	input vector
\mathbf{u}_c	-	commanded input vector
V	-	eigenvalue difference weighting parameter
\mathbf{x}	-	state vector
\mathbf{y}	-	output vector
α	-	angle of attack
δ_c	-	elevator deflection
ζ	-	damping ratio
θ	-	damping ratio angle
λ_{ach}	-	achievable pole (eigenvalue)
λ_{des}	-	desired pole (eigenvalue)
$\underline{\sigma}$	-	minimum singular value
ω_d	-	damped natural frequency
ω_n	-	natural frequency
$\det[\]$	-	determinant of a matrix
$[\]^T$	-	transpose of a matrix
$[\]^{-1}$	-	inverse of a matrix
$[\]^*$	-	hermitian matrix

Abstract

An advantage of linear quadratic regulator (LQR) design is that it gives a robust system by guaranteeing stability margins. This property is used to develop an algorithm for placing robust poles. The algorithm chooses the positive semidefinite weighing matrix \mathbf{Q} and positive definite weighting matrix \mathbf{R} by attempting to place closed loop poles at a set of desired poles. If the desired poles lie outside the allowable LQR region, the algorithm finds the achievable poles inside the region that are closest to the desired poles. The solution requires using a gradient search technique to minimize a weighted eigenvalue difference cost function. The weighting of the eigenvalue difference establishes the relative importance between the poles. In a multi-input multi-output system, the placement of one pole effects the allowable placement region of the other poles. Thus, the heavier weighted poles have precedence and are forced closer to their desired location. The algorithm is programmed to run on the software package MATLAB and the related subroutines are discussed. Several examples are included to illustrate the use of the algorithm, some of which can be solved in closed form to compare with the algorithm's solution. The results show that this technique is accurate for selecting robust poles at or close to the desired pole locations.

A LINEAR QUADRATIC REGULATOR WEIGHT SELECTION ALGORITHM FOR ROBUST POLE ASSIGNMENT

I. Introduction

Classical flight control system (FCS) design has evolved from the need to make aircraft stable and reduce pilot workload. Early designs built stability into the airframe making them less maneuverable and more susceptible to atmospheric disturbances. Before World War I, the first FCS's were simple gyroscopic autopilots which provided stability and a return to the desired attitude after disturbances [1:1]. With the increasing complexity of modern jets, modern FCS's have become an integral part of the aircraft. Modern FCS's not only provide stability and desired responses to specific inputs, but also suppress disturbances, component variations, uncertainties, and cross coupling effects [2:27].

In the preliminary design phase, one of the goals of the control engineer is to design the FCS to allow the airplane to accomplish its mission. The mission requirements can be broken down into several basic functions which are quantified by specifications. Aircraft specifications are usually governed by documents such as Mil Standard 1797 and Federal Aviation Regulations.

The dynamics of the airplane can be approximated by mathematical modeling. The resulting differential equations establish the characteristic equation of the system. To evaluate the system, the characteristic equation is factored into roots and plotted in the complex plane. These points are called poles and they describe the performance and stability characteristics of the airplane. The math model of the airplane is called the plant, and certain variables or states (such as roll rate or bank angle) of the plant output can be fed back into the plant input as shown in figure 1. Using feedback, the poles of the system can be shifted. In this fashion, the design engineer has the ability to shape the closed loop performance characteristics of the system to meet the design goals. A variety of FCS design techniques exist; this report is concerned with pole placement and the linear quadratic regulator (LQR) method.

Background

Pole placement is a powerful tool for the control system designer. In fact, a controllable system can be forced to satisfy any desired set of poles with the appropriate linear feedback [3]. Thus, the designer can shape the system characteristics to meet the design goals by obtaining a feedback matrix such that the closed loop poles approach the desired poles. In the single-input case, the feedback matrix is unique for a given set of poles. If the poles are placed exactly at the desired locations, the unique gains give fixed robustness characteristics that the designer must live with. In the multi-input case however, there is not one unique feedback gain matrix that will yield a given set of poles. In other words, different feedback matrices will give the same poles (or eigenvalues), but each one of these feedback matrices will have a different mode shape (or eigenvector) for

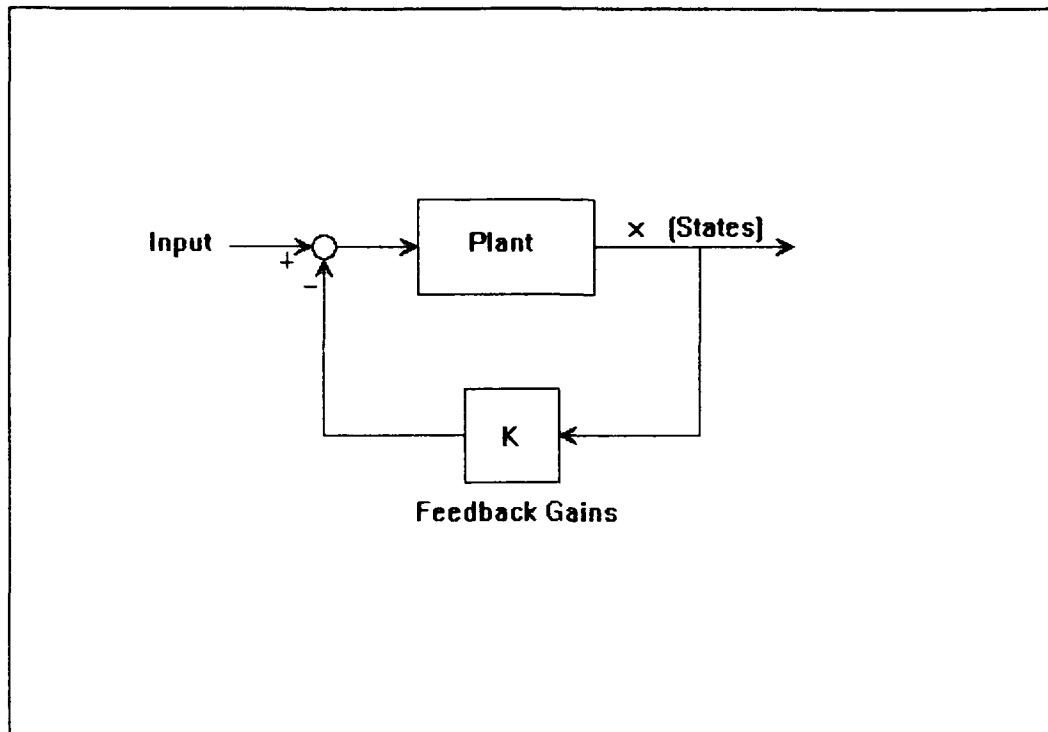


Figure 1 State Feedback Diagram

the system. For a multi-input system, Moore has shown that after the poles are placed an eigenvector set can be chosen from a class of allowable eigenvector sets [4]. This offers the freedom to first shift the poles and then choose an allowable mode shape from those available. A pole shifting technique for multi-variable feedback given by Retallack and MacFarlane shows how to select the feedback gain matrix to get any desired pole [5]. Even though their pole placement method can give the desired performance characteristics, it does not guarantee a robust system. That is, one that is insensitive to system parameter variations and external disturbances. Another way to think of robustness in a system is that it remains stable even in the case of model approximation

errors or flight condition disturbances. A typical measure of robustness in the frequency domain is the gain and phase margin.

There are pole placement methods that do achieve a robust system. However, the poles are usually restricted to a specific region of the complex plane. Simonyi and Loh developed a pole placement technique that places the closed loop poles into a region where the robustness to perturbations is determined with a condition number of the eigenvalue matrix [6]. The linear quadratic regulator (LQR) is an optimal design technique that guarantees a robust system. Anderson and Moore have shown that the LQR design guarantees robustness by giving a gain margin of $(-6, \infty)$ db and a phase margin of $(-60, 60)$ degrees [7:sect 5.4]. The difficulty lies in choosing a weighting matrix for the LQR cost function that gives the desired poles, and a great deal of research has been accomplished in this area. Kawasaki and Shimemura [8] and Wittenmark et. al. [9] have developed procedures to select weighting matrices to place poles in a specified region. Methods to place desired poles by finding the state weighting matrix are given by Graupe [10], Broussard [11], and Solheim [12]. In papers by Wilson and Cloutier [13] and Sobel and Shapiro [14], a cost function is minimized that has a trade-off between desired pole placement and desired eigenvector assignment. A method described by Harvey and Stein [15] uses the state weighting matrix to place the transmission zeros in the desired pole locations. As the gains are increased, the closed loop poles migrate towards the desired locations. In this report, an algorithm is developed that finds the LQR state weighting matrix to place a set of desired poles. The procedure varies the weighting matrices \mathbf{Q} and \mathbf{R} to find the smallest difference between an LQR achievable

pole and the desired pole. In contrast, Harvey and Stein calculate \mathbf{Q} to place the transmission zeros and vary \mathbf{R} by defining $\mathbf{R}=\rho\mathbf{I}$ and letting ρ tend towards zero. While some of their poles move to the desired locations, others escape to infinity. The algorithm given here places all the poles near their desired locations, and none go to infinity. Also, the algorithm developed in this report allows a weighting to be placed on selected poles to indicate their relative importance. A similar technique presented by Graupe finds \mathbf{Q} by minimizing the pole difference, but he does not include any weighting option or vary \mathbf{R} . In Broussard's paper, an algorithm is presented to find \mathbf{Q} by minimizing the feedback gain difference, but the gains for the desired poles must first be calculated. Broussard does introduce a weighting function in his routine, but it is used to weight the eigenvectors - not to weight selected eigenvalues. The method in this report does not calculate any eigenvectors. Instead, Moore's result stated above provides the option to select an eigenvector set after this pole placement procedure is employed. The algorithm presented here is simple to use but gives a powerful and useful technique for pole placement.

Problem Statement

Consider the state space representation of a linear time invariant system

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \quad (1)$$

$$\mathbf{y} = \mathbf{Cx} \quad (2)$$

where \mathbf{x} is an n -dimensional state vector, \mathbf{u} is an m -dimensional input vector and \mathbf{y} is an l -dimensional output vector. \mathbf{A} , \mathbf{B} , and \mathbf{C} are constant matrices of the appropriate size. Assuming (\mathbf{A},\mathbf{B}) is controllable, a linear feedback of the state variables

$$\mathbf{u} = -\mathbf{K}\mathbf{x} \quad (3)$$

can be found that shifts the closed loop poles to the desired locations. A problem arises when a control is required to drive the plant output from a nonzero state to the zero state for a plant subjected to unwanted disturbances. The problem is to drive the system back to the zero state as fast as possible while trying to use the smallest control inputs. Anderson and Moore define this as the regulator problem [7:8]. The optimal solution is to find the control input \mathbf{u} to minimize the quadratic cost function

$$J = \int_0^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \quad (4)$$

where \mathbf{Q} and \mathbf{R} are weighting matrices chosen by the designer. This is the classical linear quadratic regulator design problem. An advantage of using LQR design is that the system will always be stable and robust. The optimal solution is given by

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} \quad (5)$$

where \mathbf{P} is found from the algebraic Riccati equation

$$\mathbf{P}\mathbf{A} + \mathbf{A}^T \mathbf{P} + \mathbf{Q} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} = 0 \quad (6)$$

The LQR method guarantees robustness, but only allows pole placement in a specific region, and the poles that give the desired performance may or may not be in these regions. Therefore it may not be possible to use this method to achieve both robustness and exact pole placement. In that case a choice would have to be made. This paper takes the stance that robustness has higher priority over pole placement and hence uses the LQR technique to choose the poles. When the desired pole is outside the allowable LQR region, a gradient search is used to find the achievable pole inside the region that is the closest to the desired pole. This is done by minimizing the cost

function

$$J^* = \sum_{i=1}^n [V_i (\lambda_{des\ i} - \lambda_{ach\ i})^2] \quad (7)$$

where

- V_i is a positive definite weighting parameter for the poles,
- $\lambda_{des\ i}$ is the i^{th} desired pole, and
- $\lambda_{ach\ i}$ is the i^{th} achievable pole from LQR

Thus, when J^* is made small, the system comes close to satisfying the desired pole spectrum and is simultaneously robust.

Pole weighting is included in the cost function to give priority to poles that need to be a specific value. Some poles may have limitations that prevent deviations from the desired locations. For example, an actuator may have characteristics that determine the location of the pole, and if a designer were to move the pole he would violate the physical model. To circumvent this, a high weighting is placed on that particular eigenvalue difference in the cost function.

The LQR pole placement and the pole difference minimization functions have been compiled in an algorithm that finds the achievable poles. The algorithm is programmed into macros called MATLAB "m"-files that run on the software package MATLAB. Five test cases are run with the algorithm. First and second order systems that can be solved in closed form are compared with the algorithm poles; a third order system is run to show the effect of weighting an actuator pole; a sixth order example of an F-4 from Harvey and Stein is examined and compared to their results; and an example of an aircraft similar to

an A-4D is examined to demonstrate the effects of robust pole placement on level 1 flying qualities.

Organization

The rest of the report is contained in chapters II through V and appendices A and B. They are organized as follows:

- Chapter II presents the theory of the linear quadratic regulator method and robustness measures of single-input single-output (SISO) and multi-input multi-output (MIMO) systems. A section with definitions of observability and controllability is included to allow a complete presentation of LQR theory.
- Chapter III develops the robust pole placement algorithm. A discussion on how the algorithm is run on the software package MATLAB is included.
- Chapter IV examines five test cases and discusses the results of each. A comparison of the Harvey and Stein method of pole placement to the algorithm is presented and the limitations of the Harvey and Stein method are discussed.
- Chapter V gives the conclusions and recommendations for further study.
- Appendix A supplies a listing of the "m" files used to run the algorithm on MATLAB.
- Appendix B gives aircraft data for example five and includes the formulas for the longitudinal equations of motion.

II. Theory

Controllability and Observability [17: sect 2.2.9]

Before the linear quadratic regulator theory is introduced, some definitions that are essential to the theory are presented. A system is completely controllable if any initial state x_0 can be shifted to any final state x_f in a finite time. In other words, a feedback matrix exists that can shift a closed loop pole to any desired location. Controllability for the linear system given in equation (1) is determined by forming the controllability matrix

$$\mathbf{M}_c = [\mathbf{B} \mid \mathbf{AB} \mid \mathbf{A}^2\mathbf{B} \mid \dots \mid \mathbf{A}^{n-1}\mathbf{B}] \quad (8)$$

where n is the number of states. The system is controllable if the rank of \mathbf{M}_c is n . If the system is not full rank, $n - \text{rank}(\mathbf{M}_c)$ gives the number of poles that can not be moved by feedback.

A system is said to be completely observable if all the states x can be determined from measuring the outputs y . If a state is not observable, it does not influence the output and cannot be identified. For the linear system described by equations (1) and (2), the observability matrix

$$\mathbf{M}_o = [\mathbf{C}^T \mid \mathbf{A}^T\mathbf{C}^T \mid (\mathbf{A}^T)^2\mathbf{C}^T \mid \dots \mid (\mathbf{A}^T)^{n-1}\mathbf{C}^T] \quad (9)$$

can be formed to check if the system is completely observable. If the rank of \mathbf{M}_o equals n , the number of states, then the system is completely observable. For an unobservable system, the number of modes that can't be observed is equal to $n - \text{rank}(\mathbf{M}_o)$. For completeness two more definitions need to be given. A system is detectable if all the

unstable (right half plane) poles are observable, and a system is stabilizable if all the unstable (right half plane) poles are controllable.

Linear Quadratic Regulator [16: sect 3.3; 17: sect 6.1]

If a linear time invariant system is completely controllable, the poles can be placed anywhere in the complex plane using full state feedback. It makes sense to place all the poles in the left half plane for stability, but it is also desirable to place the poles far to the left to make the transients die out quickly. By doing this the system would converge quickly to its steady state, but the control inputs may be excessively large. The control input amplitudes are physically limited which puts a constraint on the speed of the system or how far left the poles can be shifted. The relative importance of system speed to control input amplitude naturally leads to an optimization problem.

To formulate the linear quadratic regulator problem, first consider the state space representation of a completely controllable linear time-invariant system

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (1)$$

with the full state feedback law $\mathbf{u} = -\mathbf{K}\mathbf{x}$. Then the closed loop system

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{B}\mathbf{K})\mathbf{x} \quad (10)$$

has as its poles the eigenvalues of

$$\det(\lambda\mathbf{I} - \mathbf{A} + \mathbf{B}\mathbf{K}) \quad (11)$$

which, with the appropriate feedback gains, can be shifted to any desired location. The problem is to place the poles such that the system comes to its steady state as fast as possible with the least control inputs. With this in mind, a criteria to measure the speed

in which an initial state is returned to the zero state (which is the desired steady state) is given by

$$\int_{t_0}^{t_1} \mathbf{x}^T \mathbf{Q} \mathbf{x} \, dt \quad (12)$$

where the state weighting matrix \mathbf{Q} is a positive-definite symmetric real matrix. The matrix \mathbf{Q} is chosen to indicate the relative importance of each state component. The integral in equation (12) measures the deviation of \mathbf{x} from its steady state value during the time $(t_1 - t_0)$. By minimizing this criterion, the states will now be driven to the steady state as fast as possible, but the system will experience large control amplitudes. To prevent the large control amplitudes, a term must be added to equation (12) to include the input. Thus, the criterion to be optimized is defined by

$$J = \int_{t_0}^{t_1} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) \, dt \quad (13)$$

where the controls weighting matrix \mathbf{R} is a positive-definite symmetric real matrix which determines the relative importance of the control inputs. This integral is called the linear quadratic regulator cost function or performance index. The problem now lies in finding the input \mathbf{u} that minimizes the performance index J .

From optimization theory, the feedback law $\mathbf{u} = -\mathbf{K}\mathbf{x}$ that minimizes equation (13) as $t_1 \rightarrow \infty$ is

$$\mathbf{K} = \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} \quad (14)$$

The matrix \mathbf{P} is the positive-definite solution (which is unique) of the algebraic Riccati equation

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} + \mathbf{Q} - \mathbf{P} \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^T \mathbf{P} = 0 \quad (15)$$

By requiring the matrix \mathbf{Q} to be positive definite, all the states will appear in the $\mathbf{x}^T \mathbf{Q} \mathbf{x}$

term. This ensures that all initially unstable poles are observable (which is a requirement for stability). By defining $\mathbf{H}^T \mathbf{H} = \mathbf{Q}$ and requiring $[\mathbf{A}, \mathbf{H}]$ to be detectable, the closed loop system will always be stable.

As was mentioned before, the designer has the freedom to choose the weighting matrices \mathbf{Q} and \mathbf{R} . Equation (15) shows that the choice of \mathbf{Q} and \mathbf{R} will influence \mathbf{P} , the solution of the algebraic Riccati equation. Notice from equation (14) that the matrices \mathbf{P} and \mathbf{R} play a role in determining the feedback matrix \mathbf{K} . Since the feedback matrix \mathbf{K} shifts the poles of the system, the elements of \mathbf{Q} and \mathbf{R} become the design parameters for pole placement.

LQR Stability Margins [17: chapter 7]

An important property of linear quadratic regulators is that they guarantee robust stability margins. In this section, both single-input single-output (SISO) and multi-input multi-output (MIMO) systems are shown to have guaranteed margins. But first, a relationship that is simply stated here but thoroughly derived in reference [17] is introduced. By manipulating the regulator equations (1), (14) and (15), the relationship

$$[\mathbf{I} + \mathbf{R}^{1/2} \mathbf{K}(-j\omega \mathbf{I} - \mathbf{A})^{-1} \mathbf{B} \mathbf{R}^{-1/2}]^T [\mathbf{I} + \mathbf{R}^{1/2} \mathbf{K}(j\omega \mathbf{I} - \mathbf{A})^{-1} \mathbf{B} \mathbf{R}^{-1/2}] \geq \mathbf{I} \quad (16)$$

known as the Kalman Inequality can be derived.

SISO Margins. For a SISO system, the Kalman Inequality is

$$[1 + r^{1/2} \mathbf{k}(-j\omega \mathbf{I} - \mathbf{A})^{-1} \mathbf{b} r^{-1/2}]^T [1 + r^{1/2} \mathbf{k}(j\omega \mathbf{I} - \mathbf{A})^{-1} \mathbf{b} r^{-1/2}] \geq 1 \quad (17)$$

where the lowercase bold letters are vectors. This can be simplified to

$$|1 + \mathbf{k}(j\omega \mathbf{I} - \mathbf{A})^{-1} \mathbf{b}| \geq 1 \quad (18)$$

A block diagram for the SISO system is shown in figure 2. The closed loop transfer

function of a SISO system is

$$TF_{Cl.} = \frac{G(j\omega)}{1+G(j\omega)H(j\omega)} \quad (19)$$

where $G(j\omega)=(j\omega\mathbf{I}-\mathbf{A})^{-1}\mathbf{b}$ and $H(j\omega)=\mathbf{k}$. The denominator of the transfer function for this system is known as the return difference function and is given by $1+\mathbf{k}(j\omega\mathbf{I}-\mathbf{A})^{-1}\mathbf{b}$.

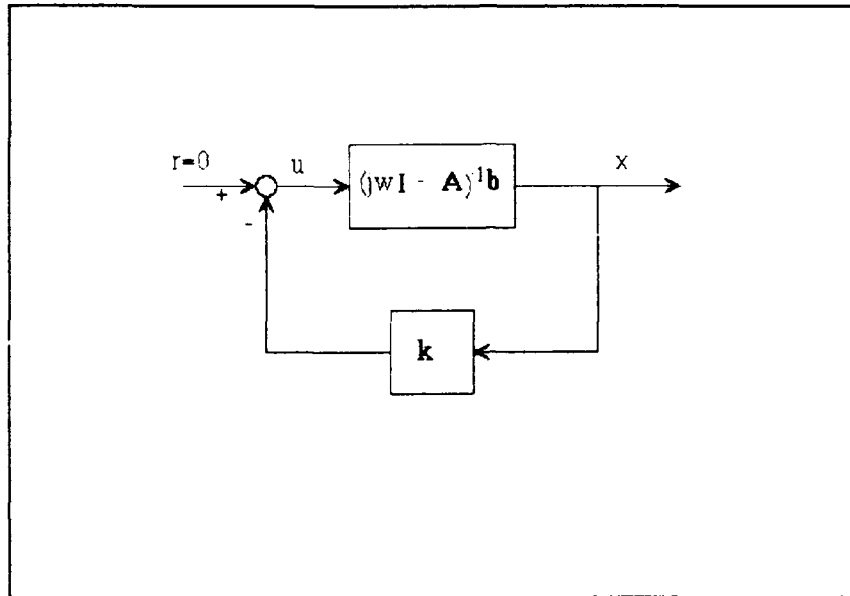


Figure 2 LQR SISO Block Diagram

The Kalman Inequality in equation (18) implies that when the open loop transfer function $\mathbf{k}(j\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{b}$ (or \mathbf{GH}) is plotted on a polar plot, the magnitude is always outside a unit circle centered at the $(-1,0)$ point. In other words, the allowable region for LQR is outside a unit disk centered at the $(-1,0)$ point. On the polar plot, the intersection of the open loop plot and a unit circle centered at the origin gives the phase margin (PM). The maximum gain margin (GM) is defined as the amount the gain can be increased before the system goes unstable (or on a polar plot, when it crosses the $(-1,0)$ point). In LQR design, the system will not go unstable because the plot cannot enter the unit circle. This

is easier to see with a picture. In figure 3 the minimum GM is at point A and the minimum PM is measured from the real axis to the lines going through points B. The GM must always be greater than $1/2$, which is inverse of the point where the polar plot crosses the real axis (or the amount the gain can be changed before the plot moves from point A to the $(-1,0)$ point). Remembering that LQR is restricted from entering the dotted unit circle shown in figure 3, the smallest PM possible is the 60° angle measured to point B. This shows that the guaranteed stability margins for LQR are

$$1/2 \leq \text{GM} < \infty \quad (20)$$

$$-60^\circ \leq \text{PM} \leq 60^\circ \quad (21)$$

MIMO Margins. In the MIMO case, the margins are first shown for $\mathbf{R}=\mathbf{I}$, and then extended to include any choice of \mathbf{R} . For a MIMO system with $\mathbf{R}=\mathbf{I}$, the Kalman Inequality reduces to

$$[\mathbf{I} + \mathbf{K}(j\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}]^*[\mathbf{I} + \mathbf{K}(j\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}] \geq \mathbf{I} \quad (22)$$

This relation holds true if and only if the minimum singular value of the return difference matrix is greater than or equal to one. That is

$$\underline{\sigma} [\mathbf{I} + \mathbf{K}(j\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}] \geq 1 \quad (23)$$

The gain and phase margins defined for the SISO case do not extend their usefulness to MIMO systems. More meaningful stability concepts called independent gain and phase margins are defined by Ridgely, Banda, and Yeh. They give the following definition:

Independent gain margins are limits within which the gains of all feedback loops may vary independently at the same time without destabilizing the system, while the phase angles remain at their nominal values. Independent phase margins are limits within which the phase angles of all loops may vary independently at the same time without destabilizing the system, while the gains remain at their nominal values [17:3-73].

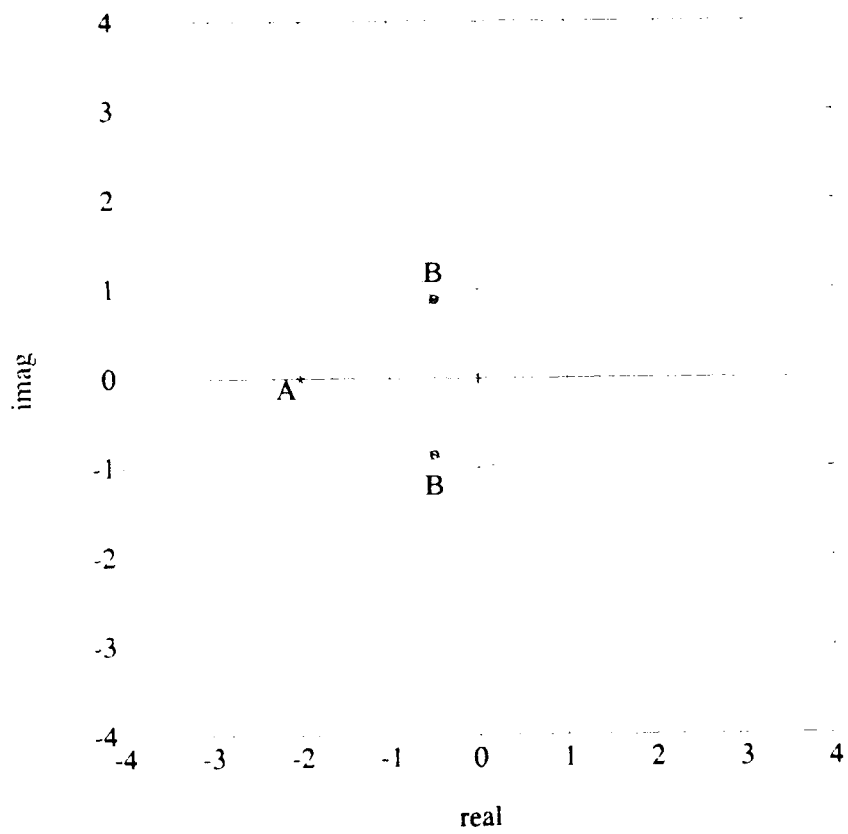


Figure 3 Polar Plot of LQR Margins

The independent gain and phase margins for a MIMO system are given by

$$\frac{1}{1+\alpha} < \text{IGM} < \frac{1}{1-\alpha} \quad (24)$$

and

$$-2 \cdot \sin^{-1} \left(\frac{\alpha}{2} \right) < \text{IPM} < 2 \cdot \sin^{-1} \left(\frac{\alpha}{2} \right) \quad (25)$$

where α = minimum singular value of the return difference. From equation (23), the smallest value of α is one. Using $\alpha=1$ in equations (24) and (25) give, as in the SISO case, guaranteed independent margins of

$$1/2 < \text{IGM} < \infty \quad (26)$$

$$-60^\circ < \text{IPM} < 60^\circ \quad (27)$$

Safonov and Athans [19] proved that these margins are the same for a general diagonal \mathbf{R} matrix for the special case when $\mathbf{Q} > 0$ and all perturbations in the various feedback loops are noninteracting. For this case, any choice of \mathbf{R} could be used since a nondiagonal matrix can always be replaced with a diagonal equivalent. If either of the two constraints are violated (which they typically are), the only form for \mathbf{R} that gives the guaranteed stability margins is $\mathbf{R} = \rho \mathbf{I}$ (where ρ is any scalar).

The guaranteed stability margin property of LQR is very desirable for the pole placement problem. However, the relationship of the weighting matrices of LQR to the location of the closed loop pole is not well understood. One method of placing poles with LQR is to vary \mathbf{Q} and \mathbf{R} by trial and error and watch the movement of the poles, but this is not very efficient. In the next section, an algorithm is presented that will find

the \mathbf{Q} and \mathbf{R} matrices that places the closed loop poles as close as possible to a set of desired poles thus simultaneously obtaining pole placement and stability robustness.

III. Development of a Robust Pole Placement Algorithm

The important properties of ensuring a system is stable and has guaranteed margins are desirable advantages of LQR. In this section, an algorithm is presented which finds appropriate LQR weighting matrices to allow the closed loop poles, placed with the LQR technique, to approach a set of desired poles. If a desired pole is outside the allowable LQR region, the algorithm finds the pole inside the LQR region (the achievable pole) that is closest to the desired pole. The algorithm also allows a weighting to be placed on the achievable poles which forces the higher weighted achievable poles closer to the desired poles. This can be an advantage in third or higher order systems, since in higher order systems the LQR region depends on the location of each pole. The algorithm tries to minimize the distance between desired and achievable poles but might not be able to place all poles exactly. If a certain pole has a heavier weighting, the algorithm tries to place it closer to the desired pole. When the weighted pole is shifted closer to its desired location the LQR achievable regions change, causing a shift in the other non-weighted poles away from their desired locations. Although the algorithm may not give the exact overall desired pole spectrum, it gives the designer the ability to enforce a required pole placement. The algorithm also delivers a system that is robust, stable, and close to the required pole specifications.

Defining the Problem

Using LQR design, the closed loop system is

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{BK})\mathbf{x} \quad (10)$$

with the optimal feedback matrix defined as

$$\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} \quad (14)$$

Using this feedback the closed loop system is

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{BR}^{-1}\mathbf{B}^T\mathbf{P})\mathbf{x} = \mathbf{A}^*\mathbf{x} \quad (28)$$

and the closed loop (achievable) poles are the eigenvalues of \mathbf{A}^* . Thus, the closed loop eigenvalues depend only on \mathbf{Q} and \mathbf{R} (since \mathbf{P} is a function of \mathbf{Q} and \mathbf{R}).

The goal of the algorithm is to minimize the cost function

$$J^* = \sum_{i=1}^n [V_i (\lambda_{des\ i} - \lambda_{ach\ i})^2] \quad (7)$$

where the pole weights V_i are chosen by the designer. In the LQR problem, the state weighting matrix is required to be positive semidefinite ($\mathbf{Q} \geq 0$) and the control weighting matrix positive definite ($\mathbf{R} > 0$). \mathbf{Q} can be made positive semidefinite by defining $\mathbf{Q} = \mathbf{H}^T\mathbf{H}$. Likewise, \mathbf{R} can be made positive definite by defining $\mathbf{R} = \mathbf{M}^T\mathbf{M}$ and requiring \mathbf{M} to be square or tall (i.e. more rows) and have full column rank. Since only a special case upholds the guaranteed stability margins for a diagonal \mathbf{R} , let \mathbf{R} be defined as $\rho\mathbf{I}$. As mentioned before, the system is always stable if the pair $[\mathbf{A}, \mathbf{H}]$ is detectable.

If \mathbf{Q} and \mathbf{R} are varied by giving small perturbations to \mathbf{H} and \mathbf{M} , the response of the pole difference cost function J^* can be determined by $\frac{\partial J^*}{\partial \mathbf{H}}$ and $\frac{\partial J^*}{\partial \mathbf{M}}$. A gradient search technique will find the minimum value of J^* when $\frac{\partial J^*}{\partial \mathbf{H}}$ and $\frac{\partial J^*}{\partial \mathbf{M}}$ is smaller than a preset tolerance.

Algorithm

The Algorithm to place robust stable poles is given by the following steps:

- 1) Define the problem by specifying **A**, **B**, and the desired pole locations.
- 2) Choose the pole weighting parameters V_i .
- 3) Provide an initial guess for **H** and ρ (since $\mathbf{R}=\rho\mathbf{I}$).
- 4) Calculate **Q** and **R**.
- 5) Solve the Riccati equation and find the gain matrix **K**.
- 6) Calculate the achievable poles.
- 7) Calculate the pole difference cost function J' .
- 8) Starting the second time through, check to see if the magnitude of $\frac{\partial J^*}{\partial \mathbf{H}}$, $\frac{\partial J^*}{\partial \mathbf{M}}$ and $|J'_k - J'_{k+1}|$ are less than a prescribed tolerance (where k is the iteration number). If so stop.
- 9) Perturb **H** and **M** and go to step 4.

Using the Algorithm With MATLAB

All the examples were run using MATLAB on a Compaq 286 computer or a VAX 11/780. MATLAB is a powerful controls software package that allows the user to write custom macros (functions) or "m" files that can be integrated with other MATLAB routines. Six new "m" files were written for the execution of the algorithm. The function FMINS is also used for the gradient search. The six "m" files are described below:

1. LQRPP - Linear Quadratic Regulator Pole Placement is the main body for the algorithm. The designer calls this function and sends the **A** and **B** matrices, the desired pole placements, and the pole weighting. It returns the achievable poles, the gain matrix

K, and the weighting matrices **Q** and **R**. It assigns the initial values to **H** and ρ (for **Q** and **R**), sets up the plotting feature, and initializes the gradient search by calling FMINS.

2. **PPFUNC** - Pole Placement Function is the function that is minimized by the gradient search routine (FMINS). This function calculates the eigenvalue difference and includes any pole weighting specified by the designer. Care is taken to direct the achievable poles toward the closest desired poles. Each achievable pole calculated during the gradient search is plotted by this function.

3. **PPMAKEH** - Pole Placement MAKE **H** is a function called by **PPFUNC** to make the symmetric **H** matrix from the vector that is being perturbed.

4. **PLOTINIT** - Plot Initialize is called from **LQRPP** to set up the plot axis size and plot the desired poles.

5. **PPINIT** - Pole Placement Initial is called by **LQRPP** to set up a vector containing the upper triangular values of an $n \times n$ identity matrix. These values are sent to **PPFUNC** to form the initial **Q** and **R** matrices.

6. **STARTPP** - Called by **LQRPP** to define global variables.

Use is made of MATLAB's gradient search routine that employs the Nelder-Mead simplex algorithm [19]. Also, MATLAB's powerful graphing capability is used to plot points of the gradient search as it converges to the achievable poles. Both initial guesses of **H** and **M** are taken to be the identity matrix **I**. All the "m" files used to run the examples in the next section are included in appendix A.

As the example problems got larger than third order, the run time on the Compaq 286 took one to several hours. In particular, the sixth order F-4 example ran for

seventeen hours before the PC ran out of memory. The VAX, however, was usually a factor of six faster than the PC; the F-4 example took only three hours to run and the third order example finished in ten minutes (clock time). On the higher order examples the plotting feature was turned off to speed up the run time.

IV. Examples

Five examples are now presented to show the effectiveness of the robust pole placement algorithm.

Example 1 - First Order SISO Case

The first order SISO is included as an example because the LQR regions and the performance index can be found in closed form. The closed form solutions allow a convenient check for the algorithm's solution.

Problem Setup. The first order SISO case can be formulated by letting the **A** matrix and **b** vector be scalars. Then the open loop state space description is

$$\dot{x} = ax + bu \quad (29)$$

Using the full state feedback $u=-kx$, the closed loop system is

$$\dot{x} = (a - bk)x \quad (30)$$

The closed loop poles or eigenvalues are

$$\det(\lambda I - a + bk) = 0 \quad (31)$$

or

$$\lambda_{cl} = a - bk \quad (32)$$

Thus, the solution to the differential equation in (30) is simply

$$x = x_0 e^{\lambda t} = x_0 e^{(a-bk)t} \quad (33)$$

where x_0 is an initial condition for x . For the optimal solution, k is found from the Riccati equation; but k can also be found in a closed form solution from the quadratic performance index. Putting equation (33) into the feedback equation gives

$$u = -kx_0 e^{(a-bk)t} \quad (34)$$

Combining equations (21) and (4), the quadratic performance index is

$$J = x_0^2 \int_0^\infty (q + rk^2) e^{2(a-bk)t} dt \quad (35)$$

$$= x_0^2 (q + rk^2) \int_0^\infty e^{2(a-bk)t} dt \quad (36)$$

$$= \frac{x_0^2 (q + rk^2)}{2(a-bk)} [e^{2(a-bk)t}]_0^\infty \quad (37)$$

$$= \frac{-x_0^2 (q + rk^2)}{2(a-bk)} \quad (k \text{ is always chosen such that } a-bk < 0) \quad (38)$$

Since all variables but k are constant, $J(k)$ can be minimized by taking the derivative with respect to k and setting it equal to zero

$$\frac{dJ}{dk} = 0 = k^2 - \left(\frac{2a}{b}\right)k - \left(\frac{q}{r}\right) \quad (39)$$

For simplicity, with no loss in generality let $b=1$. Thus, the optimal feedback gain k is

$$k_{opt} = \left[\sqrt{a^2 + \frac{q}{r}} + a \right] \quad (40)$$

For the pole placement method in this report, the control weighting matrix r is chosen to be p . In this example some insight can be gained by watching r vary. With the optimal k given by equation (40) and letting $b=1$, the optimal closed loop pole is

$$\lambda_{cl, opt} = -\sqrt{a^2 + \frac{q}{r}} \quad (41)$$

Note that the closed loop pole is always negative, so the system will always be stable.

Now let q or r in equation (41) approach infinity

$$\begin{aligned}
q \rightarrow \infty \quad \lambda_{cl\ opt} &\rightarrow \infty \\
r \rightarrow \infty \quad \lambda_{cl\ opt} &\rightarrow -|a|
\end{aligned}
\tag{42}$$

As $q \rightarrow \infty$ the system gets faster and the closed loop poles move further to the left. This demonstrates that q is indeed the weighting for the speed of response term - increase q and the system gets faster. On the other hand, the value of r determines the control effort. If control is expensive, the designer would make r big. From equation (42), as $r \rightarrow \infty$ the closed loop pole is put at the negative absolute value of a . Thus, if the open loop system is stable and $r \rightarrow \infty$, the pole does not move and no control is used. If the open loop system is unstable, the optimal solution is to shift the pole to its stable mirror image.

Equations (40) and (41) show that this example is really just a one parameter system that depends on $\frac{q}{r}$. By letting $\frac{q}{r} = \sigma$ and varying σ from zero to approximately infinity ($\sigma \neq \infty$ since $r \neq 0$, $r=0$ would signify unlimited control power), the achievable regions for the closed loop poles can be found. By looking at the open loop transfer function

$$gh_{ol} = k_{opt}(sI - a)^{-1}b = \frac{k_{opt}}{s-a} \tag{43}$$

the open loop pole is located at a . A root locus for the LQR achievable poles is drawn in figure 4 for the stable and unstable case. Observe that both closed loop systems are stable (as they had to be) and the locus can not enter the area between zero and point $-a$. That region is restricted for all closed loop poles regardless of the choice for \mathbf{Q} and \mathbf{R} . Since the cost function that was minimized was the LQR performance index, the only

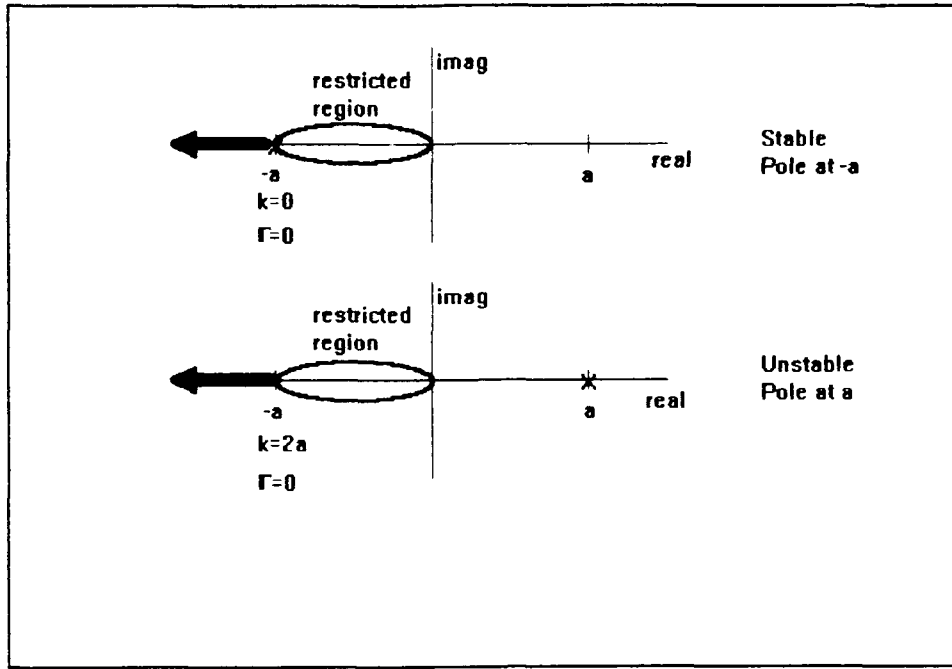


Figure 4 Root Locus for LQR Achievable Poles

allowable closed loop poles are to the left of point $-a$. To see why, look at the gain and phase margins for the system. From equation (43), the margins are calculated as

$$GM = 20 \log \left(\frac{k_{opt}}{a} \right)$$

$$PM = 180 - \tan^{-1} \left[\frac{\sqrt{k_{opt}^2 - a^2}}{-a} \right]$$

where GM is in dB and PM is in degrees. If a pole were placed in the restricted region, it would violate the guaranteed phase margin property of LQR.

Now that the allowable regions of LQR are known, consider the two cases of a stable and an unstable open loop pole. For each case, let the desired pole be

$$\lambda_{des} = a - k \tag{44}$$

so that

$$k = a - \lambda_{des} \quad (45)$$

Stable Case. Letting $a=-5$ and $\lambda_{des}=-7$ gives

$$\begin{aligned} gh_{ol} &= \frac{2}{s+5} \\ k_{opt} &= \sqrt{25+q} - 5 \\ &= a - \lambda_{des} = -5 + 7 = 2 \end{aligned} \quad (46)$$

solving for q gives

$$q = (k_{opt} + 5)^2 - 25 = 49 - 25 = 24 \quad (47)$$

Then the achievable pole is

$$\lambda_{ach} = \lambda_{CL} = -\sqrt{a^2 + q} = -\sqrt{24 + 25} = -7 \quad (48)$$

The PM and GM are guaranteed to be at least 60° and -6 dB. From the Bode plot in figure 5, the gain margin for $k = 2$ is infinite. This result implies that the system is stable and robust for any positive gain.

Unstable Case. Letting $a=5$ and $\lambda_{des}=-7$ gives

$$\begin{aligned} gh_{ol} &= \frac{12}{s-5} \\ k &= a - \lambda_{des} = 12 \end{aligned} \quad (49)$$

and

$$k_{opt} = \sqrt{25+q} + 5 \quad (50)$$

Solving for q gives

$$q = (k_{opt} - 5)^2 - 25 = 49 - 25 = 24 \quad (51)$$

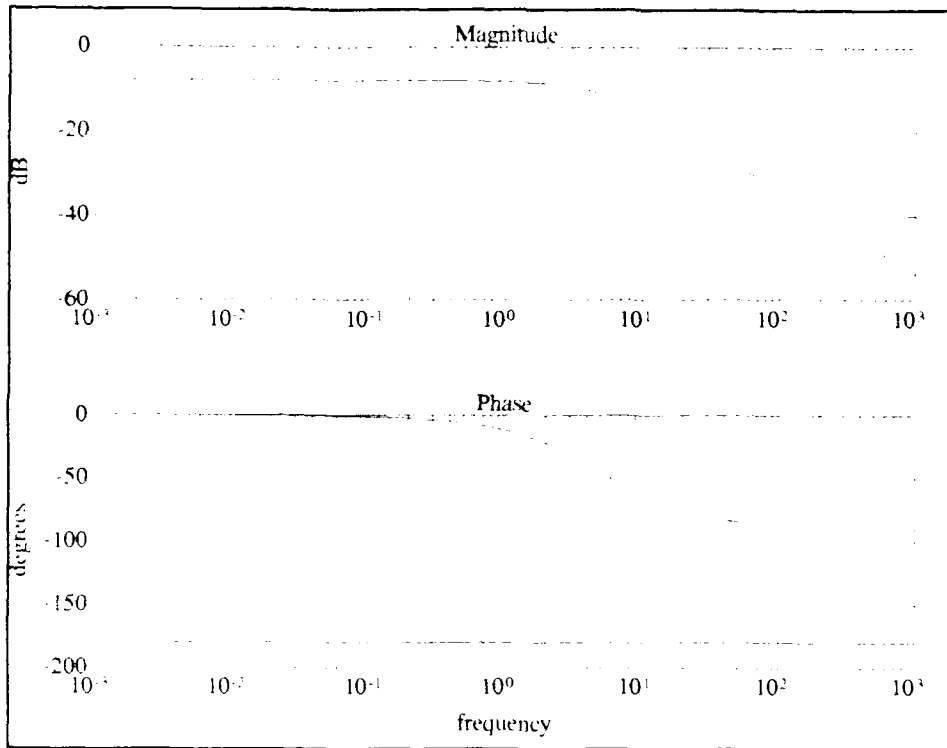


Figure 5 Bode Plot for Stable Case

then

$$\lambda_{ach} = \lambda_{CL} = -\sqrt{a^2 + q} = -\sqrt{25 + 24} = -7 \quad (52)$$

The Bode plots for this system are shown in figure 6. The GM is now $[-7.6, \infty]$ dB and the PM is 65.4 degrees, which both meet the guaranteed margins as expected. An interesting result in the unstable case is when the system uses the least control effort. That is when the unstable pole is shifted to its mirror image. Figure 7 shows the Bode plot for this case, where λ_{dec} is now at -5. The stability margins are at $[-6, \infty]$ db and 60 degrees- exactly the minimums guaranteed by LQR.

Algorithm Results. Using the results of the last section, the pole difference cost

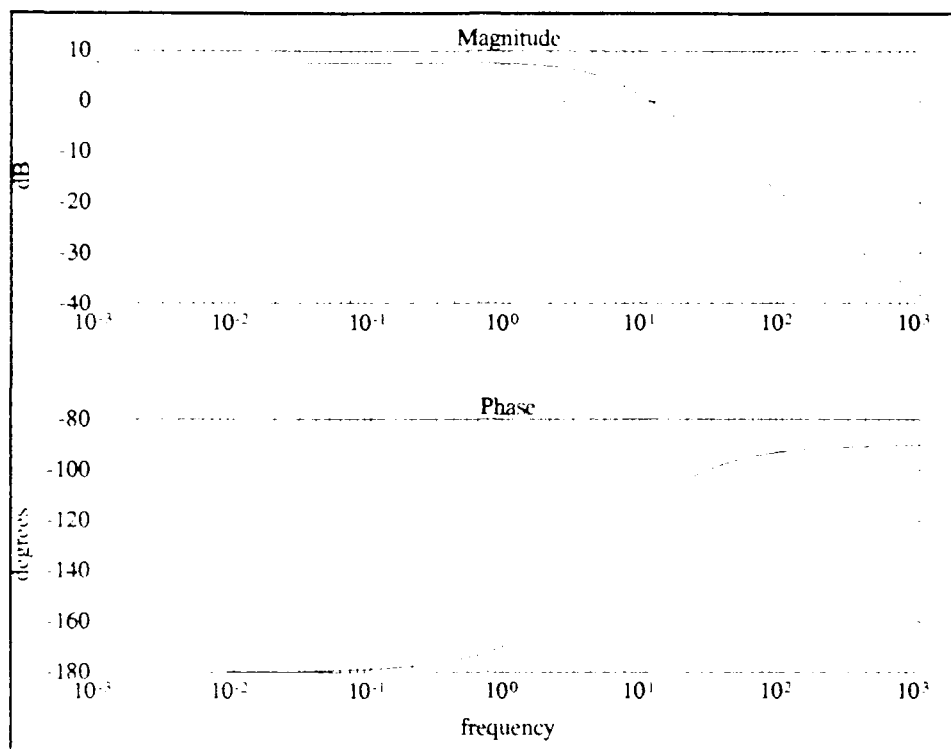


Figure 6 Bode Plot for Unstable Case $k=12$

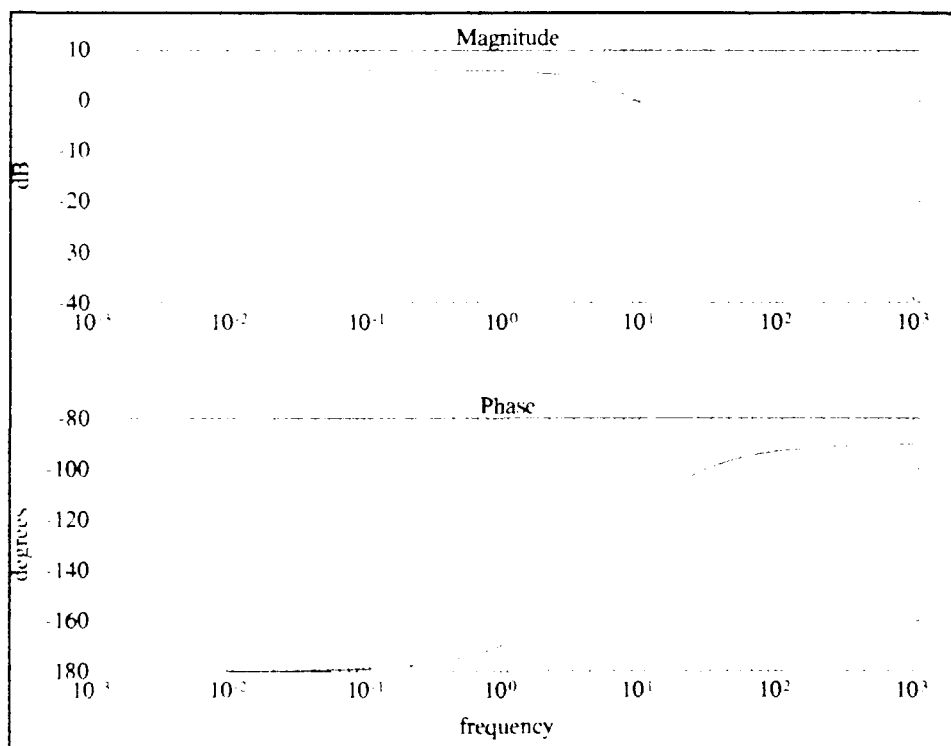


Figure 7 Bode Plot for Unstable Case $k=2a$

function can be expressed as

$$J^* = (\lambda_{des} + \sqrt{a^2 + q})^2 \quad (53)$$

where the pole weighting parameter V is not necessary since there is only one pole. Since λ_{des} and a are fixed, J^* is a function of q only. Taking the derivative with respect to q and setting it equal to zero gives the minimum. Solving for q gives

$$q = \lambda_{des}^2 - a^2 \quad (54)$$

If the desired pole is in the achievable region, the function is minimized when J^* equals zero. When the desired pole is outside the achievable region, the function is minimized when q equals zero; this will give the achievable pole location at $-\sqrt{a^2}$.

Both cases (stable and unstable) were run on MATLAB using the "m" files listed in appendix A. Figures 8 and 9 show the algorithm plot for the gradient search for the best achievable pole. The process starts with the initial value of $q=1$.

For all cases the algorithm successfully places the pole at the exact closed form location. If the desired pole were in the restricted region between zero and minus five, the achievable pole should be placed at negative five. Figure 10 shows this to be the case. Table I compares the results from the algorithm with the closed form solutions. The closed form solutions for the second unstable case were calculated the same as the other two cases.

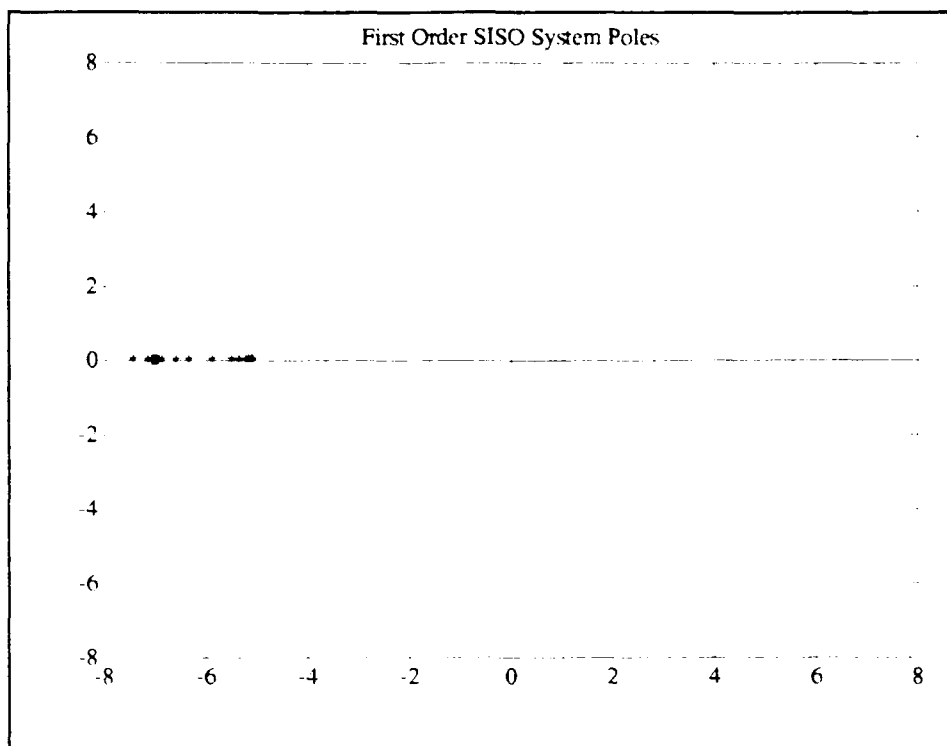


Figure 8 Algorithm Plot for Stable Case

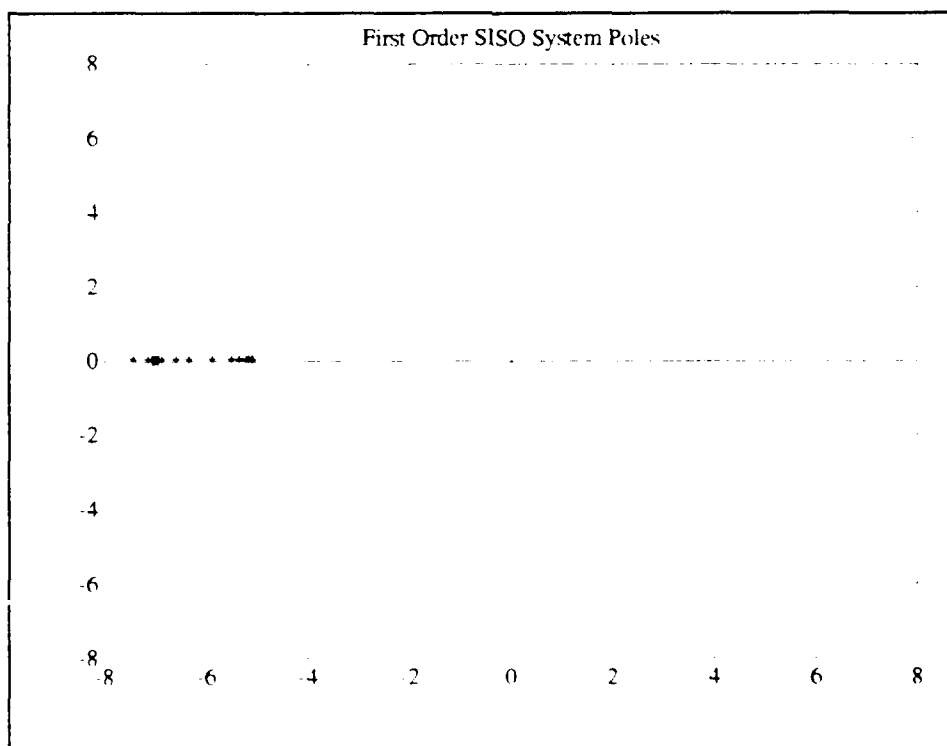


Figure 9 Algorithm Plot for Unstable Case with Desired Pole at -7

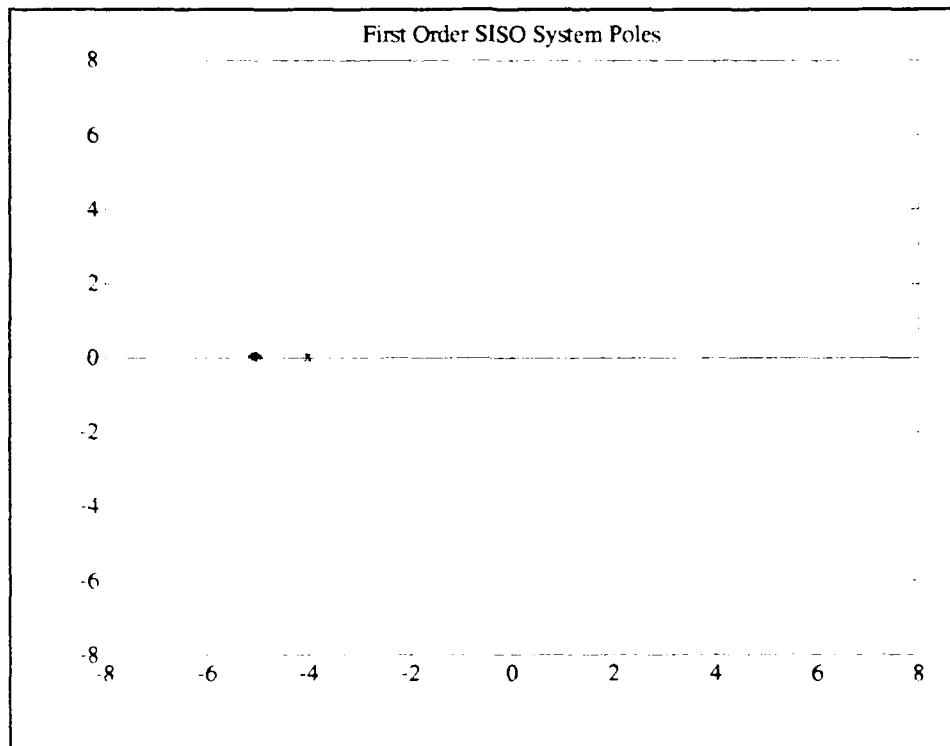


Figure 10 Algorithm Plot for Unstable Case with Desired Pole at -4

Table I First Order SISO Results

	Method	a	λ_{des}	λ_{ach}	k_{opt}	q
Stable Open Loop Pole	Algorithm	-5	-7	-7	2	24
	Calculated	-5	-7	-7	2	24
Unstable Open Loop Pole	Algorithm	5	-7	-7	12	24
	Calculated	5	-7	-7	12	24
Unstable Open Loop Pole	Algorithm	5	-4	-5	10	0
	Calculated	5	-4	-5	10	0

Example 2 - Second Order SISO Case

As the order of the problem increases, the allowable LQR regions become more complex, but with the second order case the regions can still be calculated in closed form. This allows a check for the algorithm to see if the achievable pole is indeed the closest point in the LQR region to the desired pole.

Problem Setup. The second order SISO problem can be formulated by applying a force to a frictionless mass as depicted in figure 11. The equation of motion for this system is

$$u = m\ddot{z} \quad (55)$$

By letting $x_1 = z$ and $x_2 = \dot{z}$ the state space representation is

$$\begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u \quad (56)$$

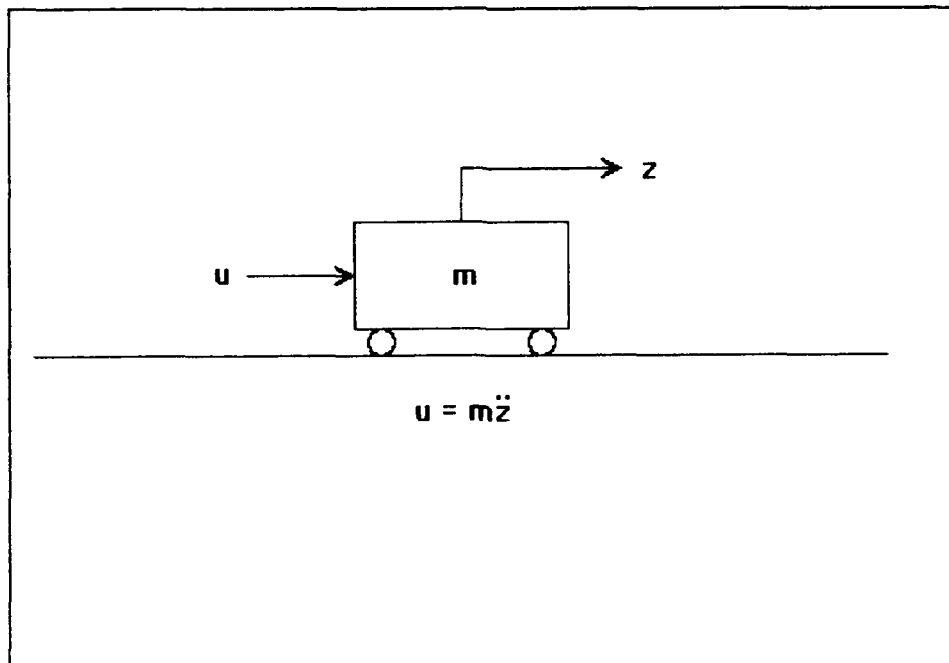


Figure 11 Second Order SISO Example

With $m=1$, the system is defined as

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u = \mathbf{A} \mathbf{x} + \mathbf{B} u \quad (57)$$

A block diagram of the system is seen in figure 12, where the full state feedback is

$$u = -[k_1 \ k_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = -\mathbf{K} \mathbf{x} \quad (58)$$

Since $u = \ddot{z}$, equation (58) can also be expressed as

$$\ddot{z} + k_2 \dot{z} + k_1 z = 0 \quad (59)$$

which is the characteristic equation for the system. To see this, look at the transfer function method for finding the characteristic equation. For the system in figure 12, the characteristic equation is

$$1 + gh = 0 \quad (60)$$

or

$$1 + \mathbf{K} [s\mathbf{I} - \mathbf{A}]^{-1} \mathbf{B} = 0 \quad (61)$$

Expanding equation (61) gives

$$s\mathbf{I} - \mathbf{A} = \begin{bmatrix} s & 0 \\ 0 & s \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} s & -1 \\ 0 & s \end{bmatrix} \quad (62)$$

$$[s\mathbf{I} - \mathbf{A}]^{-1} = \frac{\begin{bmatrix} s & 1 \\ 0 & s \end{bmatrix}}{s^2} \quad (63)$$

$$[s\mathbf{I} - \mathbf{A}]^{-1} \mathbf{B} = \begin{bmatrix} \frac{1}{s} & \frac{1}{s^2} \\ 0 & \frac{1}{s} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{s^2} \\ \frac{1}{s} \end{bmatrix} \quad (64)$$

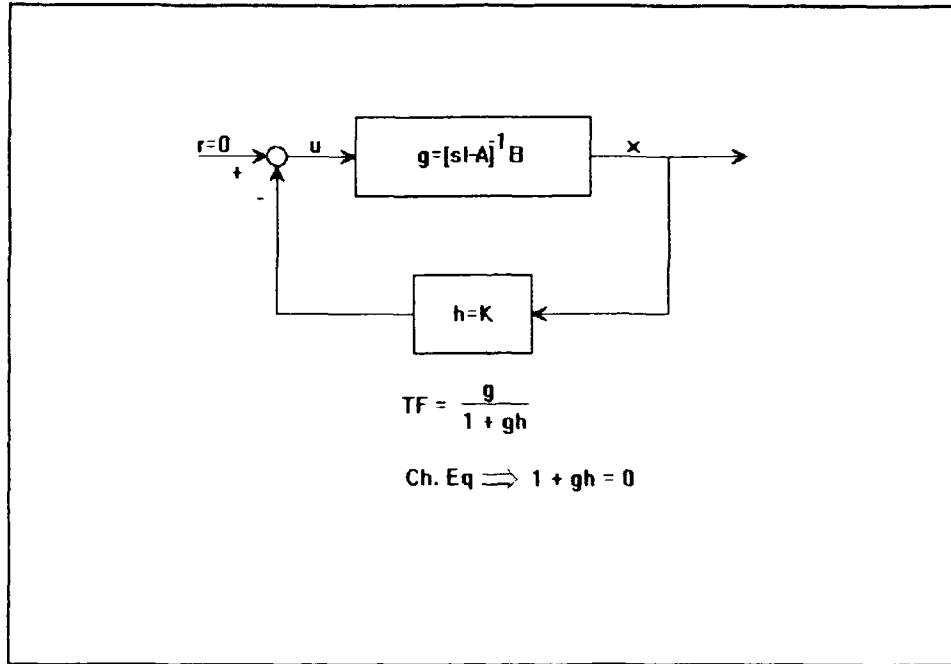


Figure 12 Block Diagram for Second Order SISO

$$\mathbf{K}[s\mathbf{I} - \mathbf{A}]^{-1} \mathbf{B} = [k_1 \ k_2] \begin{bmatrix} \frac{1}{s^2} \\ \frac{1}{s} \end{bmatrix} = \frac{k_1}{s^2} + \frac{k_2}{s} \quad (65)$$

So the characteristic equation is (from $1+gh=0$)

$$1 + \frac{k_1}{s^2} + \frac{k_2}{s} = 0 \quad (66)$$

or

$$s^2 + k_2 s + k_1 = 0 \quad (67)$$

which is the same as equation (59) expressed in the Laplace domain.

A typical second order system has the form

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = 0 \quad (68)$$

where ζ is the damping ratio and ω_n is the system's natural frequency. This example can

be made to look like equation (68) by defining k_1 and k_2 as

$$k_1 = \omega_n^2 \quad k_2 = 2\zeta\omega_n \quad (69)$$

The open loop system (gh) is $\mathbf{K}[s\mathbf{I} - \mathbf{A}]^{-1}\mathbf{B}$, or

$$gh = \frac{k_2 s + k_1}{s^2} \quad (70)$$

Using k_1 and k_2 from equation (69) and shifting to the frequency domain gives

$$gh(j\omega) = \frac{2\zeta\omega_n j\omega + \omega_n^2}{-\omega^2} \quad (71)$$

Dividing through by ω_n^2 to normalize the problem and defining $\bar{\omega} = \frac{\omega}{\omega_n}$ results in

$$gh = \frac{1 + 2\zeta\bar{\omega}j}{-\bar{\omega}^2} \quad (72)$$

Normalizing the problem this way will show that the optimization problem is not a function of frequency (ω_n). The open loop system can be represented in phaser form as

$$gh(j\bar{\omega}) = \frac{\sqrt{1 + 4\zeta^2\bar{\omega}^2}}{\bar{\omega}^2} \angle(\tan^{-1} 2\zeta\bar{\omega} - 180 \text{ deg}) \quad (73)$$

To find the PM for this system, set the magnitude equal to one ($|gh|=1$) and solve for the frequency. Choosing the real solution for the frequency gives

$$\bar{\omega} = \sqrt{2\zeta^2 + (4\zeta^4 + 1)^{\frac{1}{2}}} \quad (74)$$

Putting this value into the angle term of equation (73) and subtracting from -180 degrees gives the PM

$$\text{PM} = \tan^{-1} 2\zeta \left[2\zeta^2 + (4\zeta^4 + 1)^{\frac{1}{2}} \right]^{\frac{1}{2}} \quad (75)$$

By normalizing the frequency, the system has been reduced to a one parameter problem.

Finding the LQR Regions. On a Nyquist plot, LQR design prohibits the locus of

points of $GH(j\omega)$ from entering a unit circle centered at $(-1,0)$. In figure 3, if the plot of $GH(j\omega)$ were exactly on the dotted unit circle (centered at $(-1,0)$) LQR would have its minimum stability margins. From vector addition, the radius of the dotted unit circle in figure 3 is $1+GH$. Thus, a mathematical constraint on a Nyquist plot from LQR design is

$$|(1 + GH)| \geq 1 \quad (76)$$

In this second order example,

$$1 + gh = \frac{s^2 + k_2 s + k_1}{s^2} = \frac{s^2 + 2\zeta\omega_n s + \omega_n^2}{s^2} \quad (77)$$

And in the normalized frequency domain

$$(1 + gh) = \frac{1 + 2\zeta\bar{\omega}j - \bar{\omega}^2}{-\bar{\omega}^2} \quad (78)$$

Putting equation (78) into the constraint equation (76) leads to

$$\bar{\omega}^4 + (4\zeta^2 - 2)\bar{\omega}^2 + 1 \geq \bar{\omega}^4 \quad (79)$$

or

$$(4\zeta^2 - 2)\bar{\omega}^2 + 1 \geq 0 \quad (80)$$

This constraint will hold true for any frequency only if

$$4\zeta^2 - 2 \geq 0 \quad (81)$$

Thus, the damping ratio must meet

$$\zeta \geq \frac{1}{\sqrt{2}} = 0.7071 \quad (82)$$

The relationship of ζ and ω_n to poles in the complex plane is depicted in figure 13.

Using the relation $\cos^{-1}\zeta = \theta$, where θ is measured as in figure 13, the achievable region for LQR design is where $\theta \leq 45^\circ$. This result is plotted in figure 14. Now that the LQR

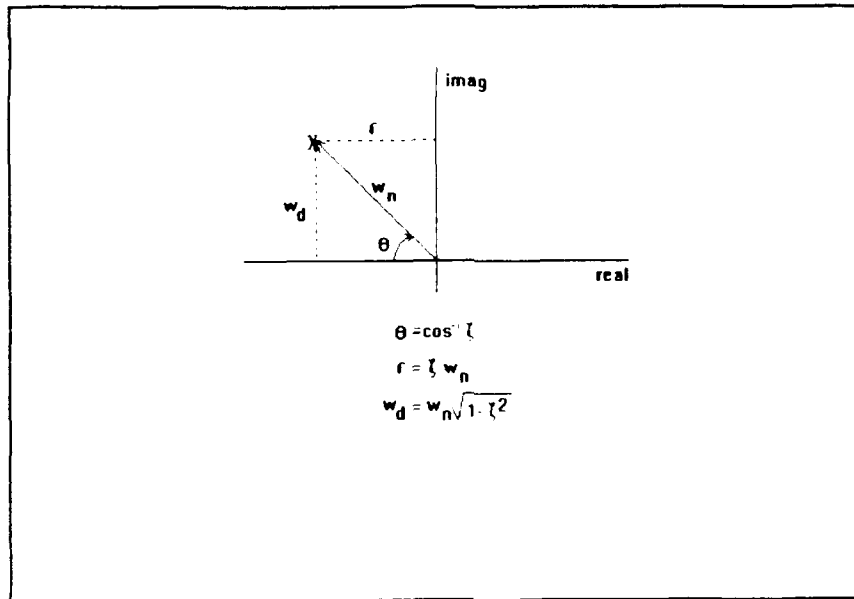


Figure 13 Relationship of Pole to Parameters

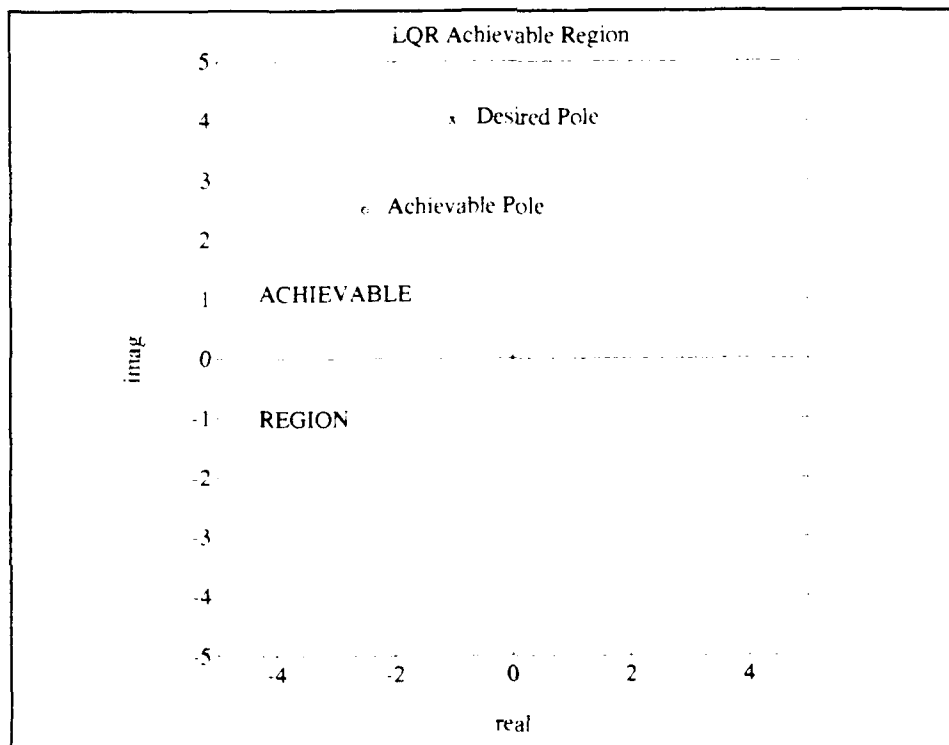


Figure 14 LQR Barriers

achievable region is known, the desired poles need to be chosen. Consider the following system specifications:

- peak overshoot $M_p \leq 0.456$
- rise time $t_r \leq 0.44$ seconds
- settling time $t_s \leq 4.6$ seconds
- robust system

Then, to just meet the system performance requirements, $\omega_n = 4.12$ and $\zeta = 0.2425$ [20:sect 2.7.4]. From the diagram in figure 13, the desired pole is found to be at $\lambda_{des} = -1 + 4j$. The desired pole meets the first three specifications, but has a PM of only 27° . To find the best achievable pole (one that is robust and closest to the desired pole), the LQR barrier must be examined. Using the top half of the plot in figure 14, the equation of the LQR barrier is

$$y = \frac{-\sqrt{1 - \frac{1}{2}}}{\frac{1}{\sqrt{2}}} x \Rightarrow y = -x \quad (83)$$

The point in the LQR region that is closest to the desired pole is at the intersection of the barrier and a line perpendicular to the barrier passing through λ_{des} . Thus, the equation of the line perpendicular to the barrier and passing through $-1+4j$ is

$$y = x + 5 \quad (84)$$

The intersection of equations (83) and (84) yields the achievable pole, or

$$\lambda_{ach} = -2.5 + 2.5j \quad (85)$$

The achievable pole gives the following system characteristics:

- $M_p = 0.04$

- $t_r = 0.51$ seconds
- $t_s = 1.84$ seconds
- $\zeta = 0.7071$
- $\omega_n = 3.54$
- $PM = 65.5^\circ$

The Phase Margin was calculated with equation (75) and can be verified with the Bode plot in figure 15. The Nyquist plot in figure 16 shows that the system does stay outside the unit circle centered at $(-1,0)$.

Notice that the PM is not exactly 60° . The LQR design guarantees only a minimum PM of 60° , but larger margins are often achieved. That is precisely the case in this example. To show why LQR can not produce a PM of 60° for this system,

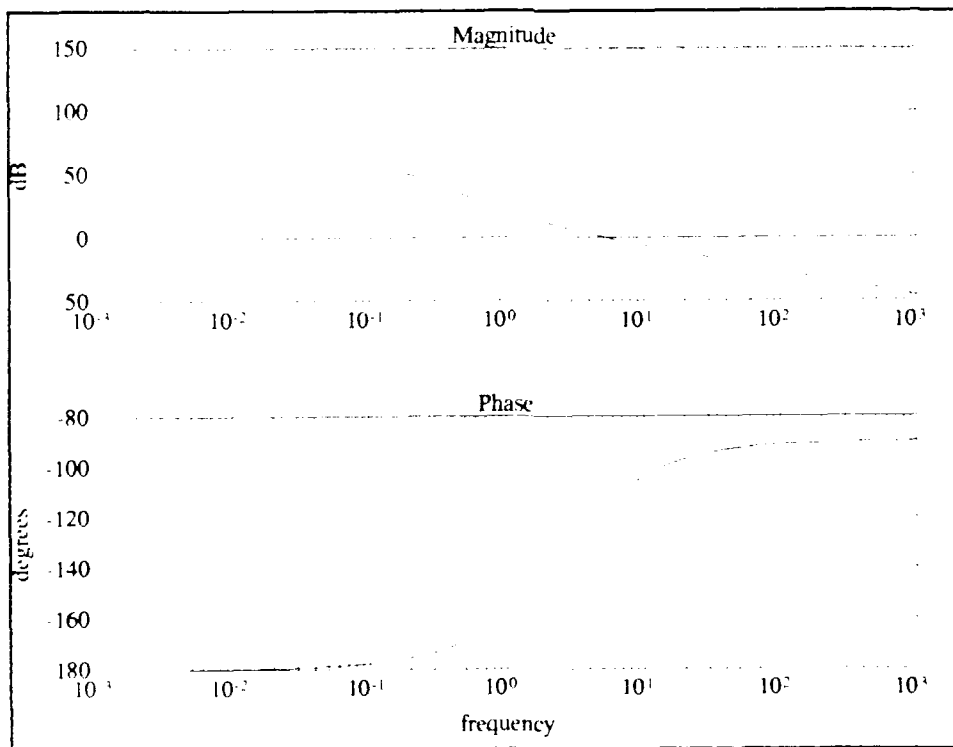


Figure 15 Bode Plot for Second Order SISO System

equation (75) was solved for ζ (with $PM=60^\circ$). The value $\zeta=0.612$ was found which corresponds to the barrier angle $\theta=52.2^\circ$. Using the same analysis of dropping a perpendicular line from the desired pole to the barrier yields $\lambda = -2.3 + 2.98j$ for a $PM=60^\circ$. This gives $\omega_n=3.76$ which allows the gains and a Nyquist plot to be calculated. The Nyquist plots for the LQR system and the $PM=60^\circ$ system are shown in figure 17. Notice that the $PM=60^\circ$ plot violates the unit circle and thus can not be generated by LQR design.

Algorithm Results. The algorithm was run on MATLAB and generated the plot in figure 18. In the plot, x marks the desired pole while o shows the achievable pole. Notice that the points calculated during the gradient search do not move outside the 45° lines. Table II gives the results for this example. The gains for the closed form solution were calculated with equation (69). The results show that the algorithm is able to accurately find λ_{ach} . The MATLAB "m"-file used to solve the second order SISO system is included in appendix A. Run time on the Compaq 286 was 4 minutes.

Table II Second Order SISO Results

	λ_{ach}	K	Q
Algorithm	-2.5 + 2.5j -2.5 - 2.5j	[12.5 5]	$\begin{bmatrix} 156.2 & 0 \\ 0 & 0 \end{bmatrix}$
Calculated	-2.5 + 2.5j -2.5 - 2.5j	[12.5 5]	not calculated

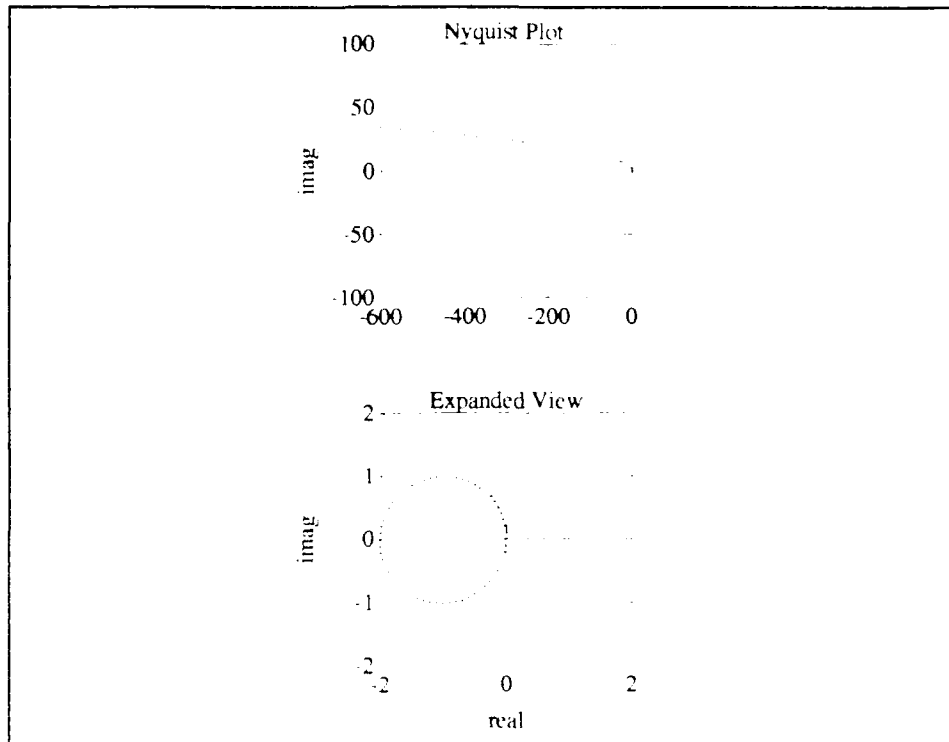


Figure 16 Nyquist Plot for LQR Design - Second Order System

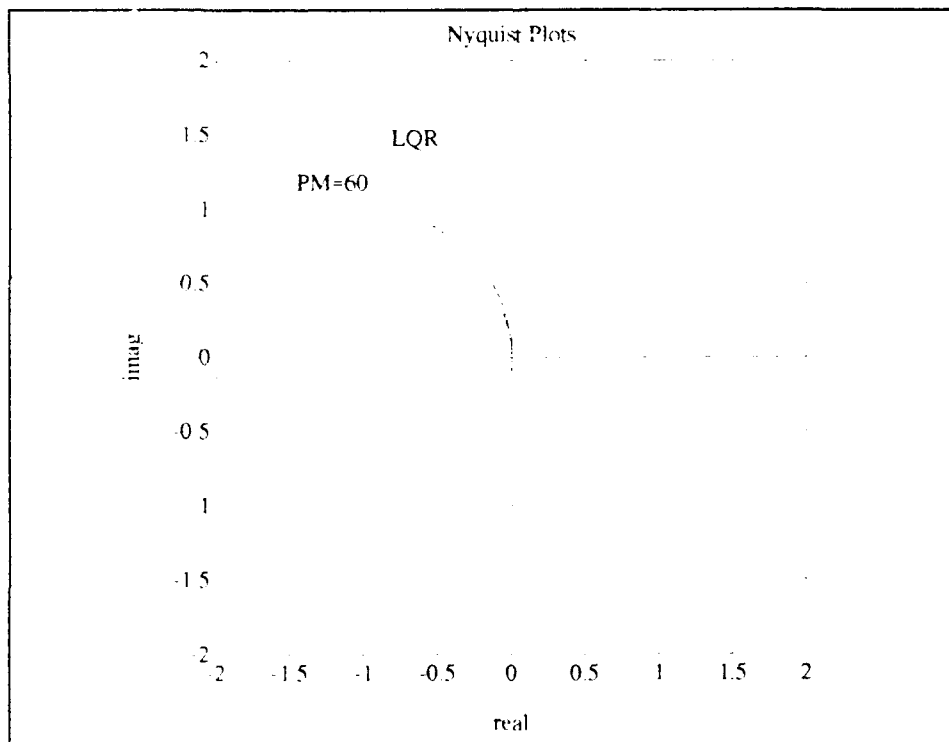


Figure 17 Comparison of LQR to 60° PM plot

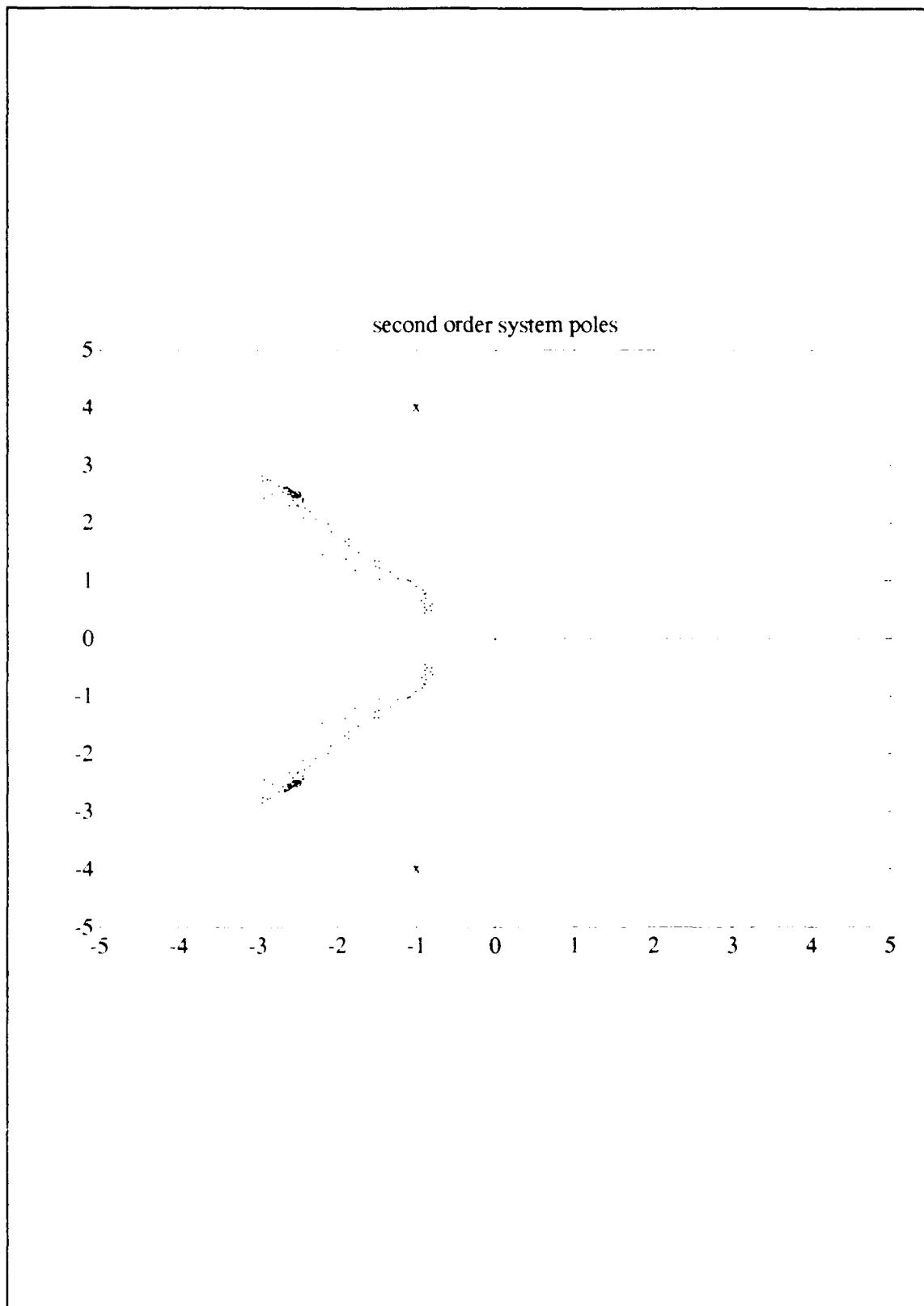


Figure 18 Algorithm Plot With LQR Barriers

Example 3 - Third Order SISO Case

In the second order example the LQR barriers could be found in closed form since the solution depended only upon ζ . The third order system cannot be reduced to a single parameter and solved in closed form, but the results of the previous sections show that the technique used is accurate. Three cases are presented in this third order example; the first chooses desired poles inside the LQR region, the second shows the effects of weighting an actuator pole and the effects on the other closed loop poles, and the third case examines the effects on a pole that is both outside the allowable LQR region and close to the imaginary axis.

Problem Setup. If an actuator is added to the second order problem it becomes a third order system. The system can be modeled as in figure 19. Letting $m=1$ and defining the states as

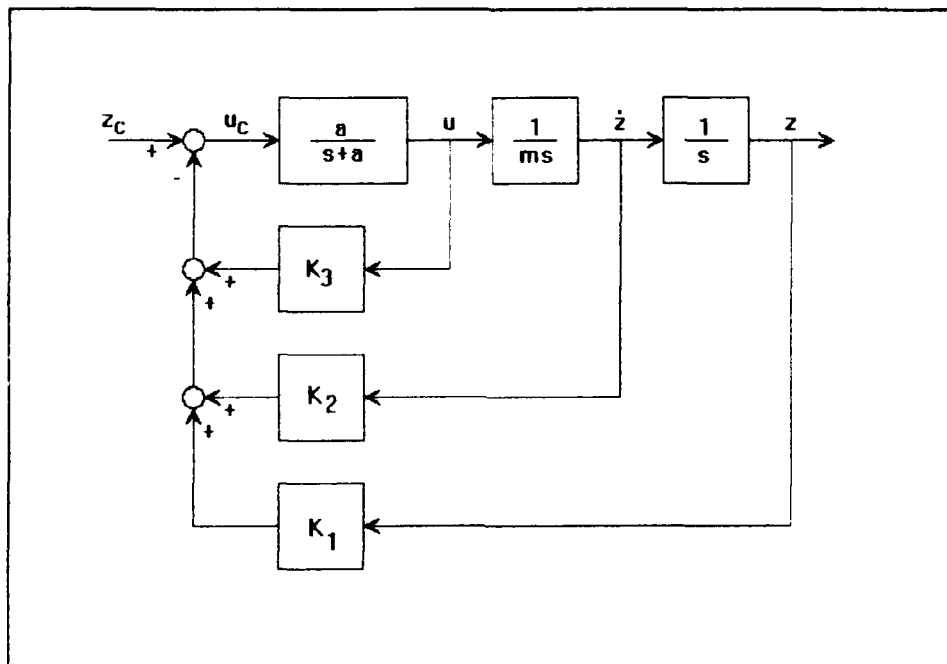


Figure 19 Third Order System Block Diagram

$$x = \begin{Bmatrix} z \\ \dot{z} \\ u \end{Bmatrix} \quad \dot{x} = \begin{Bmatrix} \dot{z} \\ \ddot{z} \\ \dot{u} \end{Bmatrix} \quad \text{where } \dot{u} = (-u + u_c)a \quad \text{and} \quad \ddot{z} = \frac{u}{m}$$

the state space representation is

$$\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -a \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} u_c \quad (86)$$

$$u_c = [k_1 \ k_2 \ k_3] x \quad (87)$$

The actuator pole is set by physical constraints, but for the sake of this example let $a=1$.

Then, from $1 + \mathbf{K}[s\mathbf{I} - \mathbf{A}]^{-1}\mathbf{B}$, the characteristic equation is

$$s^3 + (1 + k_3)s^2 + k_2s + k_1 = 0 \quad (88)$$

Factoring the characteristic equation gives three roots, one from the actuator and two from the plant.

Case 1. For this case the desired poles should be within the LQR achievable region. In the second order example, the barriers made 45° lines with the real axis. Since the third order LQR allowable regions are unknown, choose a desired location within the second order system region. Let

$$\lambda_{des} = [-0.5 + 0.5j \quad -0.5 - 0.5j \quad -1] \quad (89)$$

Using the function **PLACE** on **MATLAB**, the closed loop poles can be placed in the desired locations and the gain matrix \mathbf{K} calculated. The resulting gains are

$$\mathbf{K} = [0.5 \quad 1.5 \quad 1.0] \quad (90)$$

With these gains, the characteristic equation is

$$s^3 + 2s^2 + 1.5s + 0.5 = 0 \quad (91)$$

whose roots are the same as λ_{des} . From the Bode plot in figure 20, the PM=65.5° and the GM=∞. The system is within the limits guaranteed by LQR design, so the algorithm should place the poles exactly. The algorithm generated the plot in figure 21 and did achieve the desired pole location. To achieve the desired poles, the Q and R are

$$R = 8.293 \quad \text{and} \quad Q = \begin{bmatrix} 2.073 & 0.829 & -0.003 \\ 0.829 & 2.067 & -0.006 \\ -0.003 & -0.006 & 0.0 \end{bmatrix}$$

Case 2. In this case, the desired poles are chosen outside of the LQR allowable region. The first run is done with no weighting on the actuator pole, and the second run weights the actuator pole three times as heavily as the other poles. The desired poles need to lie outside the 45° second order system barrier. Thus, let

$$\lambda_{des} = [-3+5j \quad -3-5j \quad -10] \quad (92)$$

Using the function PLACE on MATLAB gives

$$K = [34 \quad 9.4 \quad 0.6] \quad (93)$$

For these gains, the Bode plot in figure 22 shows the PM=53.34°. The algorithm was run on MATLAB to find the LQR achievable poles and plotted in figure 23. The x's denote the desired poles, the dots are the achievable pole choices generated during the algorithm's gradient search, and the + 's are the actuator pole choices from the algorithm. It can be seen that the desired poles were outside the LQR allowable region. The achievable poles and phase margin are

$$\lambda_{ach} = [-3.48 - 4.52i \quad -3.48 + 4.52i \quad -10.78] \quad (94)$$

$$PM = 62.45^\circ \quad (95)$$

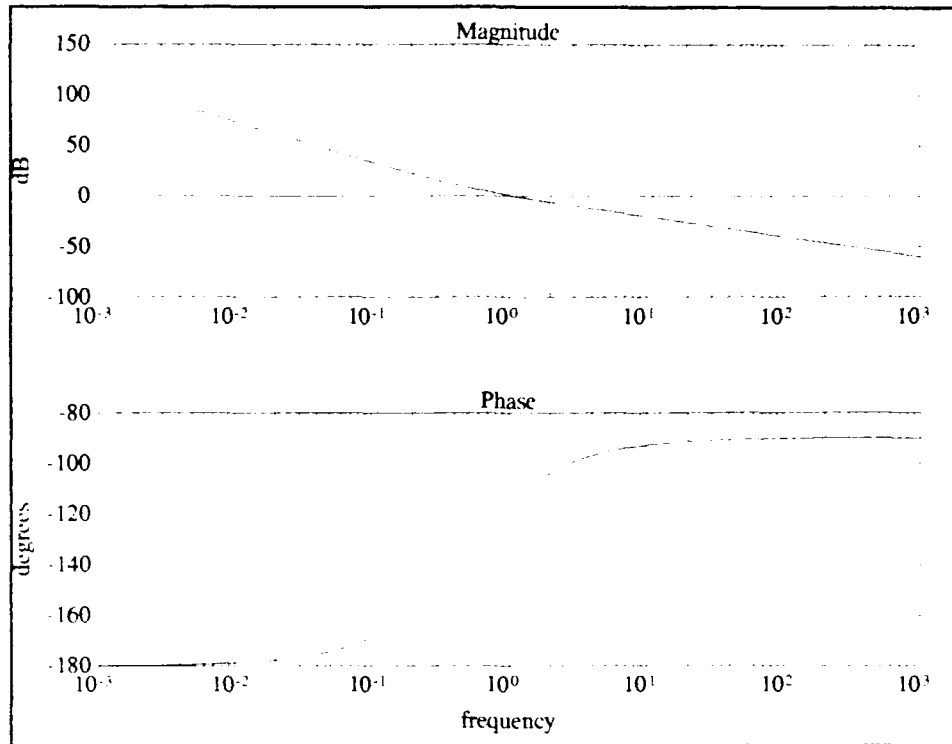


Figure 20 Bode Plot for Desired Poles - Case 1

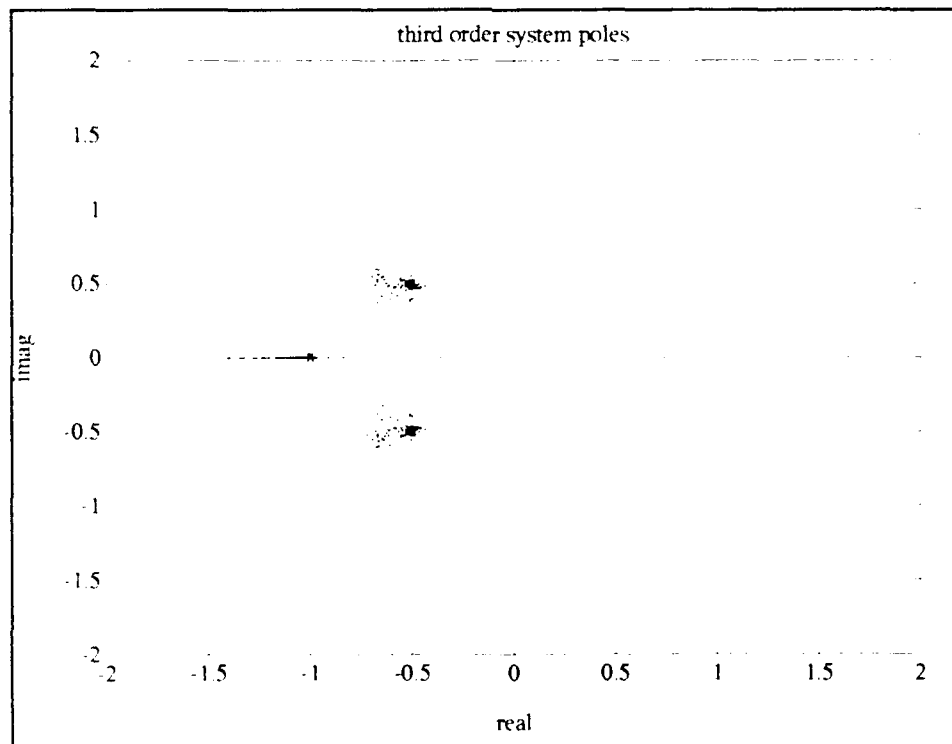


Figure 21 Algorithm Plot - Case 1

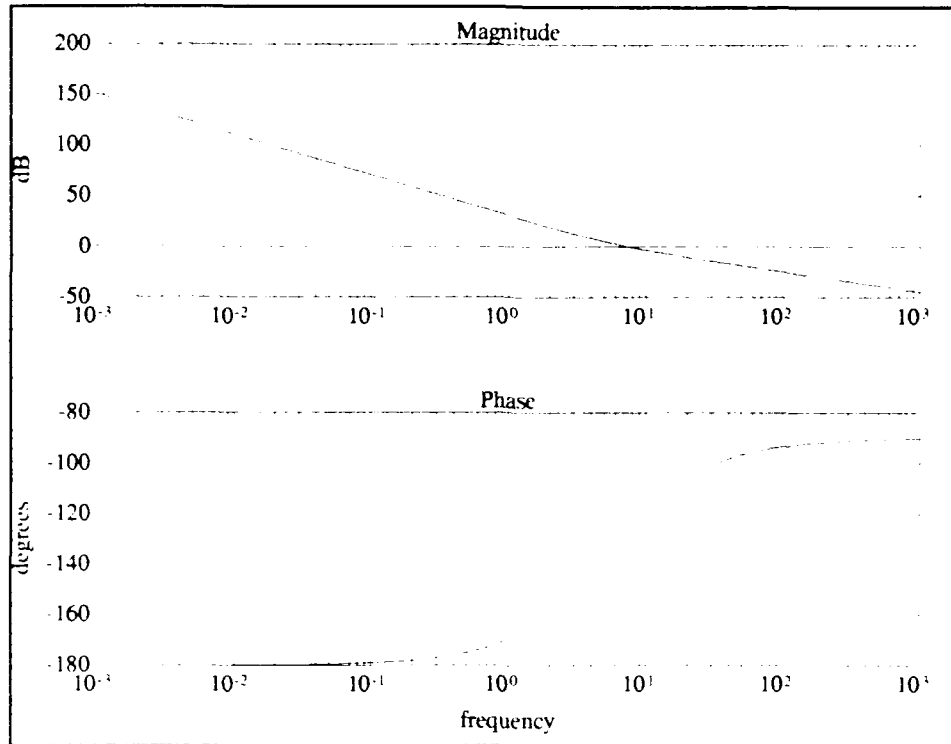


Figure 22 Bode Plot for Desired Poles - Case 2

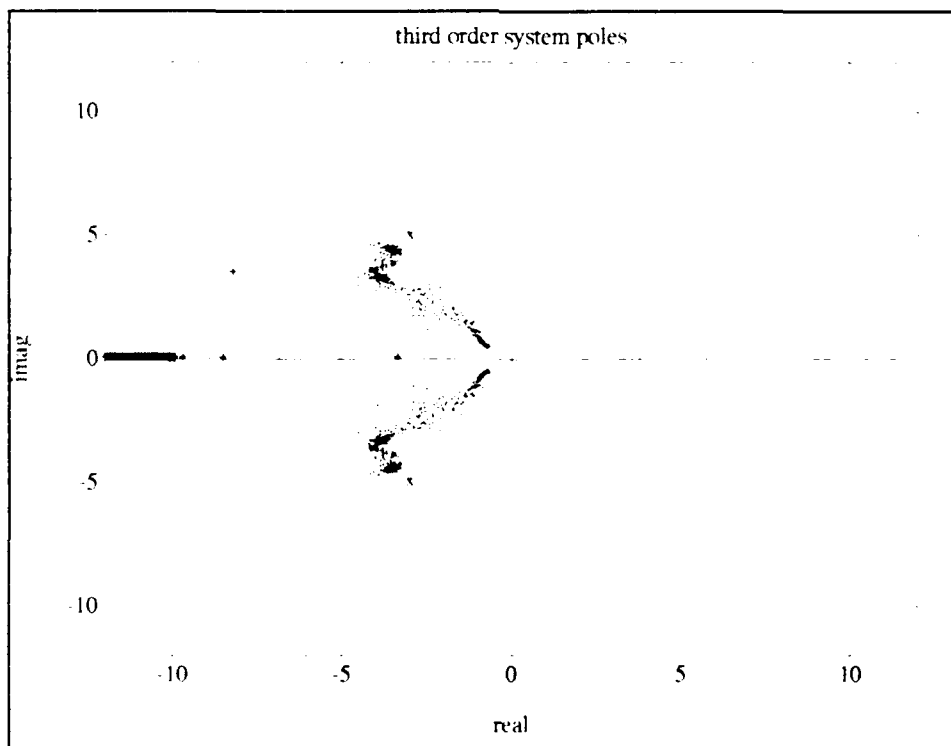


Figure 23 Algorithm Plot - Case 2 (no Weighting on Actuator)

To see the effects of weighting a pole, the same desired poles were used but the actuator pole was weighted three times as heavily as the plant poles. The algorithm found the best LQR achievable poles as

$$\lambda_{ach} = [-3.62 + 4.30i \quad -3.62 - 4.30i \quad -10.53]$$

The gradient search from the algorithm is plotted in figure 24. Again, the x's are the desired poles and the dot and plus symbols denote the gradient search attempt to find the best achievable pole. All achievable pole choices on the plot are within the LQR allowable region, and the pattern of the achievable pole choices gives an outline for the LQR barriers. However, the LQR regions depend on the location of all the closed loop poles, so the algorithm plot can not give a true estimate of the LQR barriers. Using the gains found by the algorithm, the Bode plot shown in figure 25 was generated.

Table III lists the results of the three different runs for case two. Notice from Table III that the Phase Margin increased 14.5% for the non-weighted actuator condition and 16.3% for the weighted actuator condition. The weighted actuator pole moved 2.5% closer to the desired location compared to the non-weighted actuator pole, but at the expense of the other poles. When the actuator pole was weighted, the other two plant poles were 1.5% less in magnitude (ω_n) and 4.8% less in damping angle (θ) from the non-weighted locations. There is a trade off in pole location accuracy, and the designer must decide if the benefits gained from the weighted pole outweigh the penalties to the remaining poles.

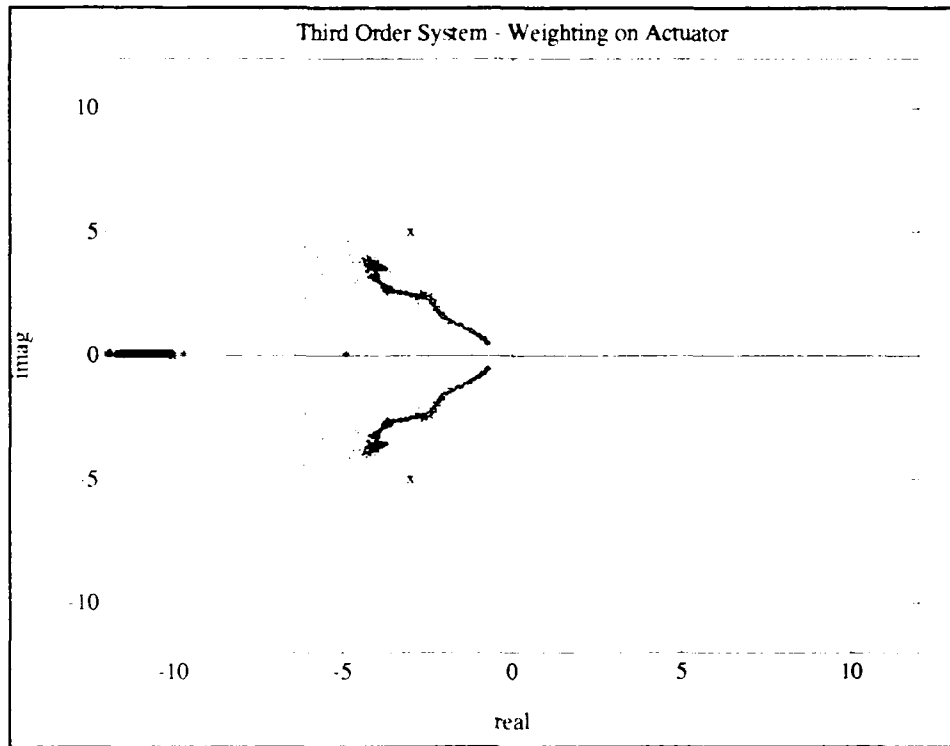


Figure 24 Algorithm Plot - Case 2 (Actuator Weighted)

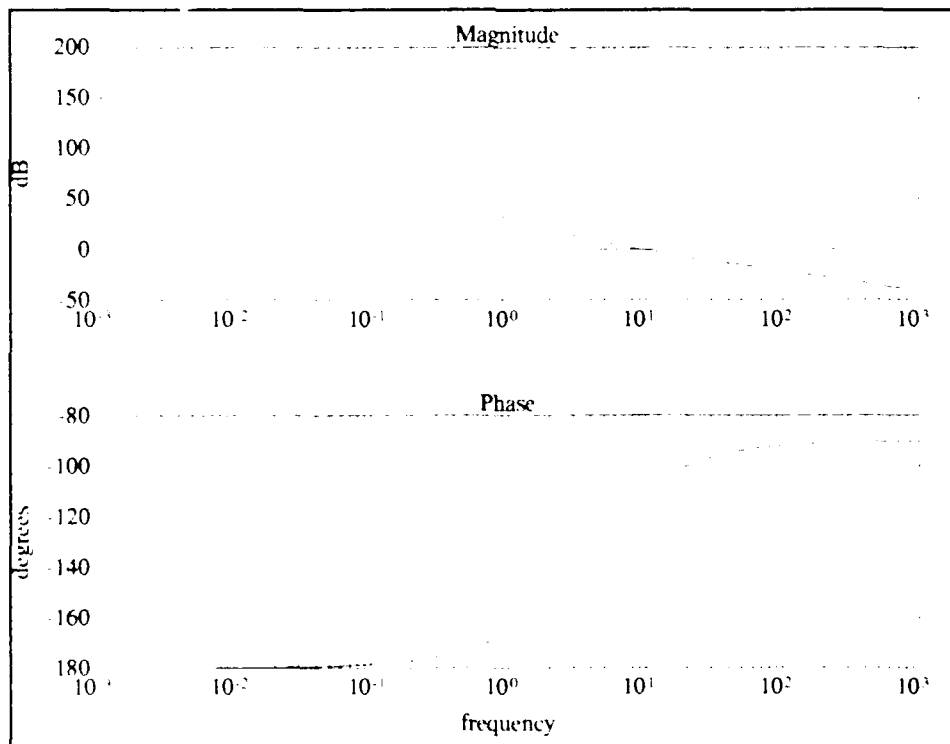


Figure 25 Bode Plot for Achievable Poles - Case 2 (Actuator Weighted)

Table III Third Order System Case 2 Results

	Pole Location	Gains	PM
Desired Pole Location	$-3 \pm 5i$ -10	[34 9.4 0.6]	53.34
No Weighting on Poles	$-3.48 \pm 4.52i$ -10.78	[35.35 10.82 0.78]	62.45
Actuator weighting	$-3.62 \pm 4.30i$ -10.53	[33.29 10.79 0.78]	63.75

Case 3. Like case 2, the plant poles are chosen outside the LQR achievable regions. They are also chosen with poor PM to demonstrate a large increase in PM with a small shift in position. The actuator pole is chosen at a location of -2.5. The desired plant poles are $\lambda_{des} = [-.2 \pm .75i]$. Using the MATLAB function `PLACE`, the gain matrix for the desired poles was found. This allowed a Bode plot and the PM to be found. The Bode plot for this system is seen in figure 26. Running the algorithm to locate the achievable poles produced the plot in figure 27. In this case, the **Q** and **R** matrices from the achievable poles in case two were used to reduce the run time. The Bode plot for the achievable poles is in figure 28. Using the function `MARGIN` on MATLAB the PM was found to be 60.74°. This can also be seen from the Bode plot in figure 28. The results are compared in table IV. Notice that for a 6.2% penalty in magnitude (ω_n) and a 24.5% smaller damping angle (θ) there is a 110% increase in PM. Figure 29 shows the location of the plant desired and achievable poles.

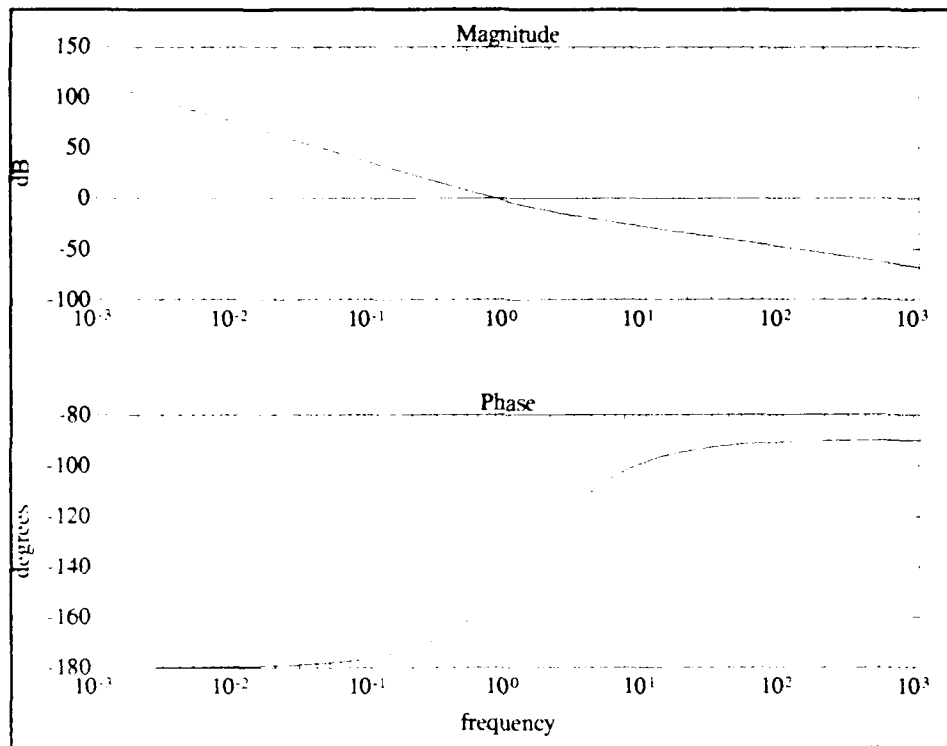


Figure 26 Bode Plot for Desired Poles - Case 3

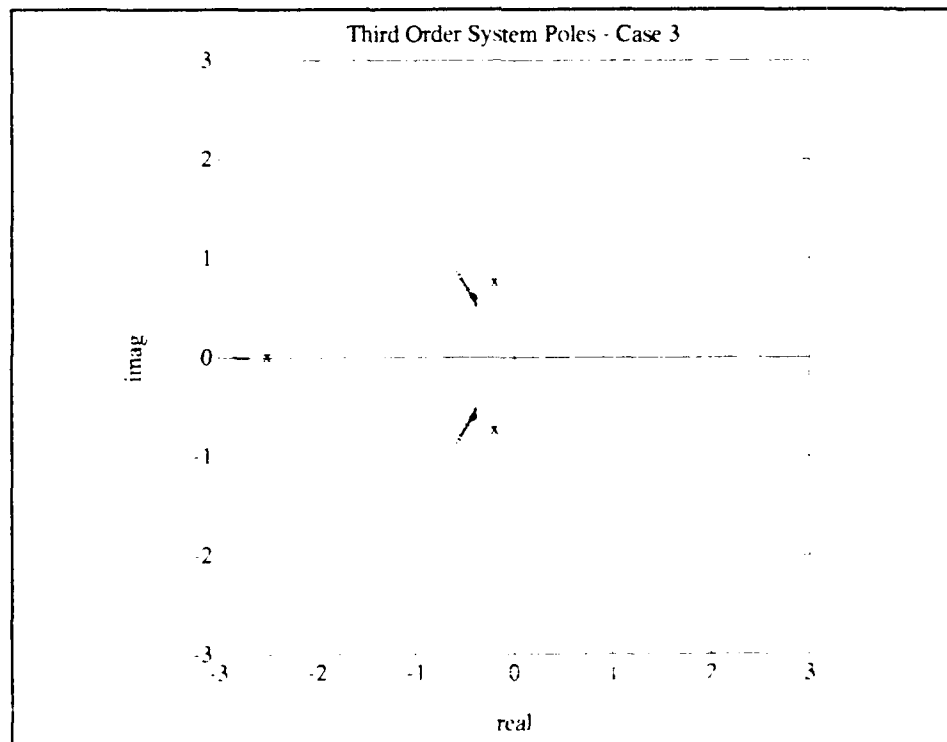


Figure 27 Algorithm Plot - Case 3

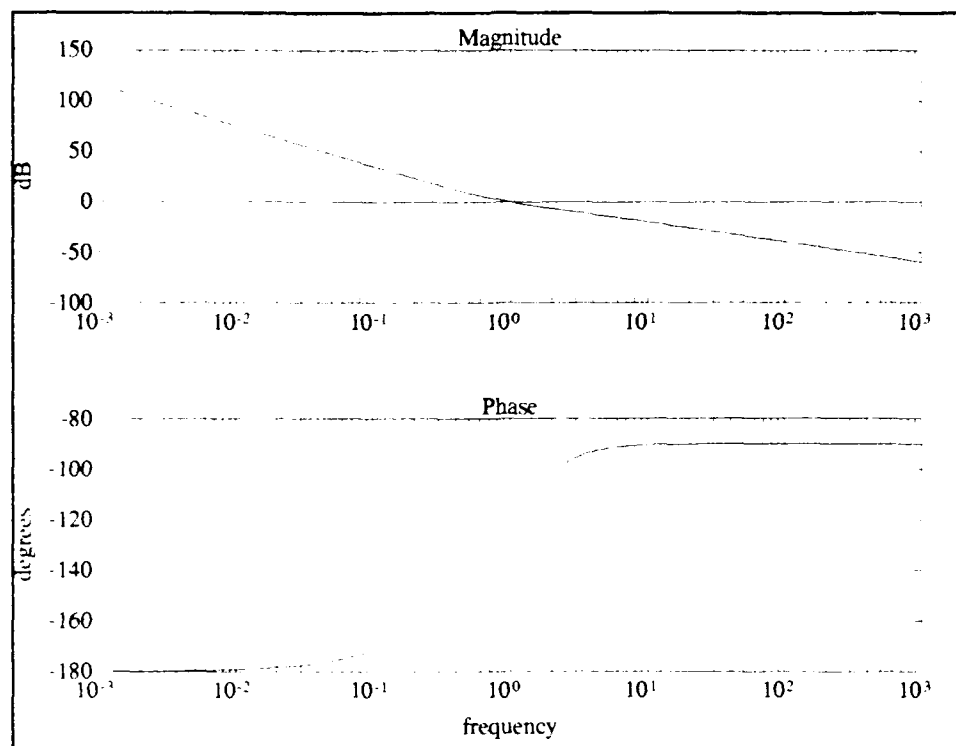


Figure 28 Bode Plot for Achievable Poles - Case 3

Table IV Third Order System Case 3 Results

	Pole Location	PM
λ_{des}	-0.2 - 0.75i -0.2 + 0.75i -2.5	28.90°
λ_{ach}	-0.4 - 0.61i -0.4 + 0.61i -2.77	60.74°

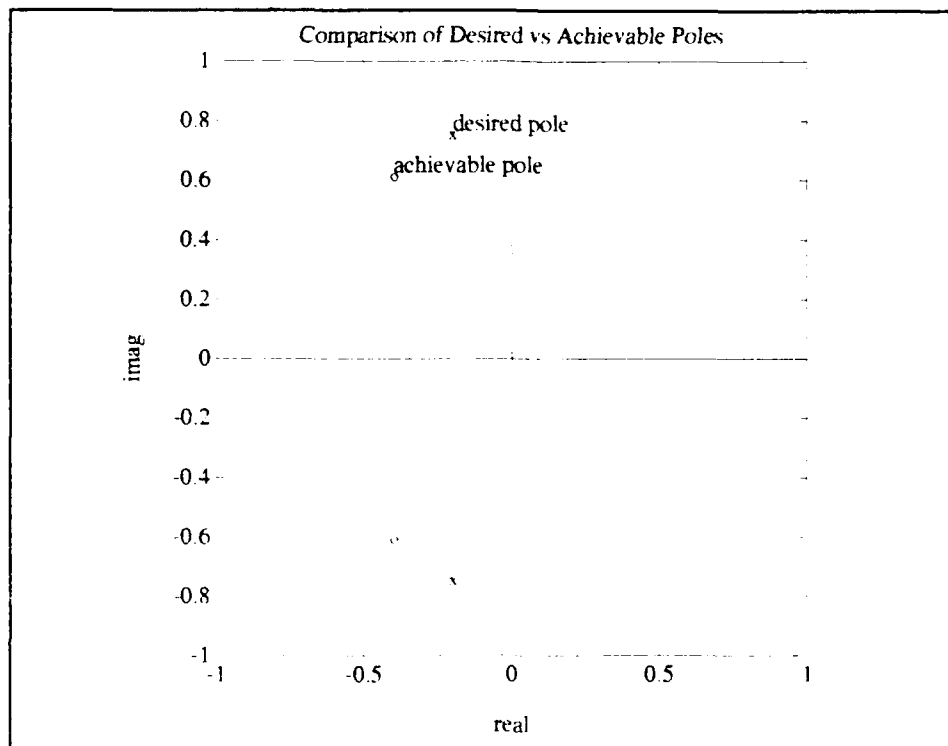


Figure 29 Comparison of Desired and Achievable Poles

Example 4 - Comparison with Harvey and Stein method [15]

Harvey and Stein present a procedure to asymptotically place eigenvalues and eigenvectors using optimal feedback. Their technique is to formulate the state weighting matrix Q such that the transmission zeros of the system are located at the desired pole locations. Then, as the control weights tend to zero (or the feedback gains increase to infinity), the closed loop poles migrate towards the desired pole locations.

This technique has one major fault; the number of transmission zeros available to be placed is $n-m$ (# states - # control inputs). As some of the closed loop poles migrate towards the desired locations, others head off to infinity. The poles that head off to infinity happen to be the actuator poles. Harvey and Stein's solution to this problem is

to monitor the actuator poles as the gains are increased and choose the gains that put the actuator poles at their required locations. They use an F-4 aircraft inner loop lateral axis design example to demonstrate their procedure. The states and controls are

$$x = \begin{bmatrix} p_s \\ r_s \\ \beta \\ \phi \\ \delta_r \\ \delta_a \end{bmatrix} \begin{array}{l} \text{stability axis roll rate} \\ \text{stability axis yaw rate} \\ \text{angle of sideslip} \\ \text{bank angle} \\ \text{rudder deflection} \\ \text{aileron deflection} \end{array} \quad (96)$$

$$u = \begin{bmatrix} \delta_{rc} \\ \delta_{ac} \end{bmatrix} \begin{array}{l} \text{rudder command} \\ \text{aileron command} \end{array} \quad (97)$$

which gives

$$\dot{x} = \begin{bmatrix} -.746 & .387 & -12.9 & 0 & .952 & 6.05 \\ .024 & -.174 & 4.31 & 0 & -1.76 & -.416 \\ .006 & -.999 & -.0578 & .0369 & .0092 & -.0012 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -20 & 0 \\ 0 & 0 & 0 & 0 & 0 & -10 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 20 & 0 \\ 0 & 10 \end{bmatrix} u$$

or

$$\dot{x} = A x + B u$$

From Mil Spec 8785B, the desired poles are

- Roll subsidence mode -4.0
- Dutch roll mode -0.63 ± 2.42j
- Spiral mode -0.05

and the actuators lie at

- Rudder actuator -20
- Aileron actuator -10

Two points must be made about the Harvey and Stein example:

1. The control weighting is defined as $\rho \mathbf{R}$ and the procedure is to let ρ tend towards zero. But to allow the actuator poles to achieve the desired locations, the procedure has to be iterated for many different values of ρ . In their example, Harvey and Stein found that a value of $\rho=0.0025$ put the actuator poles close to the desired locations. As ρ goes to zero the plant poles go to the desired locations. But the actuator pole position determines the value of ρ and hence the closed loop plant pole locations.
2. Since the actuator poles head out to infinity as the gains are increased (or ρ is decreased), Harvey and Stein modified the initial pole locations in the \mathbf{A} matrix. The open loop actuator poles were moved to (-10, -5) to allow freedom to increase the gains before the actuator poles moved past the desired locations. There has to be some play in the gains to allow the other poles to migrate towards the transmission zeros, and Harvey and Stein modified their example to start the actuator poles to the right of their desired locations.

When the F-4 data was run with the algorithm presented in this report, the \mathbf{A} matrix was changed back to the original actuator values (-20, -10) to reflect the correct locations for the actuator poles. This example was run on the VAX using MATLAB. Both the algorithm and Harvey and Stein were able to place all the poles close to their desired locations. Figures 30 and 31 show the desired and achievable pole locations for

both the algorithm results and Harvey and Stein's solution. Table V compares the results for this example. Notice that Harvey and Stein achieved poles close to the desired plant locations, but the algorithm was 12% closer to the desired aileron actuator pole and 4% closer to the desired rudder actuator pole. The robustness for a MIMO system can be measured using singular values as discussed in chapter II. Both Harvey and Stein's solution and the algorithm's solution were examined for robustness using singular values, and the singular value plots for these two cases are shown in figure 32. Two robustness tests can be checked using singular values. By defining $\mathbf{K}^*[\mathbf{sI}-\mathbf{A}]^{-1}\mathbf{B}=\mathbf{T}^*$ (where the * depends on which case is being tested), the tests are

$$\begin{aligned}\underline{\sigma}[\mathbf{I} + \mathbf{T}^*] &\geq 0 \text{ dB} \\ \underline{\sigma}[\mathbf{I} + \mathbf{T}^{*-1}] &> -6 \text{ dB}\end{aligned}\tag{98}$$

The first test is plotted in figures 33 and 34, and the second test is plotted in figures 35 and 36.

Table V Example Four Results

	Harvey and Stein	Algorithm
λ_{ach}	-3.810 -0.049 -0.727 \pm 2.358j -10.43 -22.44	-3.998 -0.091 -0.669 \pm 2.365j -10.025 -20.053
\mathbf{K}	$\begin{bmatrix} .132 & .882 & -1.576 & -.026 & -.681 & .026 \\ -.524 & -.42 & 2.827 & -.021 & .013 & -.86 \end{bmatrix}$	$\begin{bmatrix} -.306 & -1.389 & .729 & .039 & .107 & -.089 \\ .409 & .858 & -.060 & .035 & -.044 & .239 \end{bmatrix}$
PM	$[-53^\circ, 53^\circ]$	$[-60^\circ, 60^\circ]$
GM	$[-5.56 \text{ dB}, 19.7 \text{ dB}]$	$[-6.02 \text{ dB}, \infty \text{ dB}]$

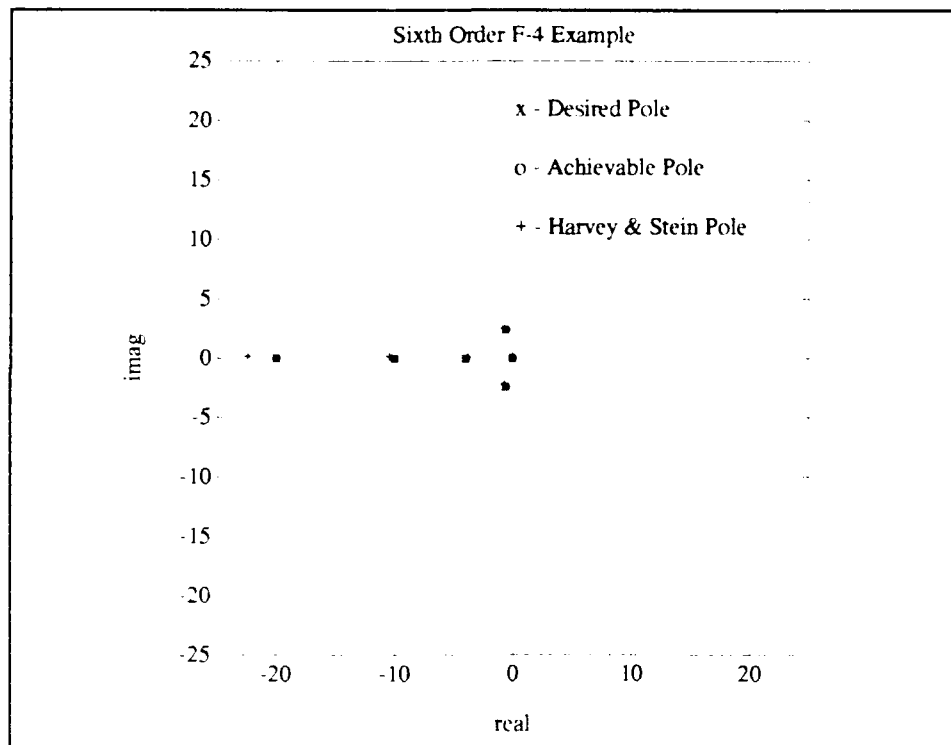


Figure 30 Comparison of Desired and Achievable Poles

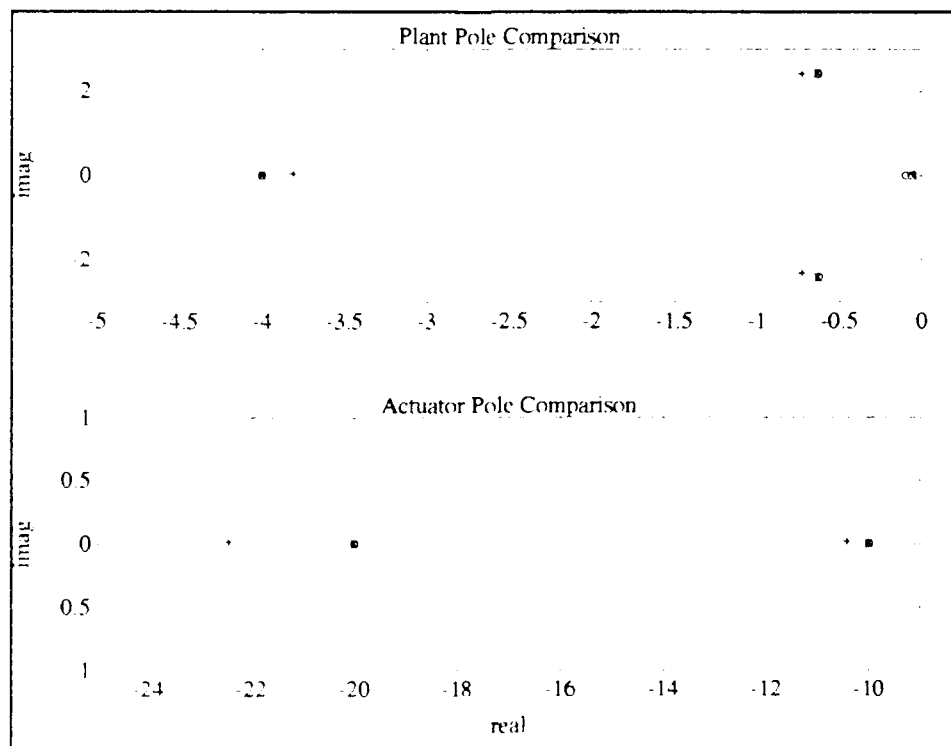


Figure 31 Desired and Actuator Poles - Expanded View

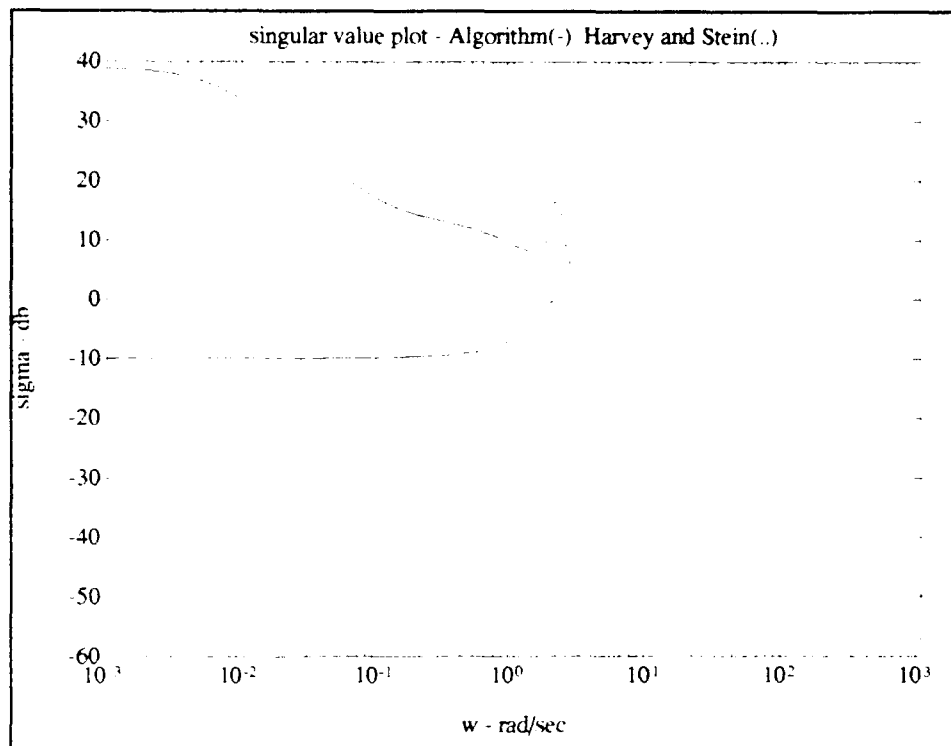


Figure 32 Singular Value Plots for Example 4

Using the minimum singular values from figures 33 and 34, the independent gain and phase margins were calculated with equations (24) and (25).

As was pointed out in chapter II, \mathbf{R} can be diagonal for guaranteed stability margins only for a special case. That happens when the disturbance matrix is diagonal or each one is independent of the others. If the disturbances are not all independent, the restriction is even tighter; \mathbf{R} must equal $\rho \mathbf{I}$ for guaranteed margins. In this example, $\rho=25.38$. When \mathbf{R} is allowed to be diagonal, where the diagonal elements do not equal each other, the guaranteed robustness breaks down. This proves that the disturbances in the various feedback loops interact with each other [19]. A plot of robustness test 1 for the diagonal \mathbf{R} matrix

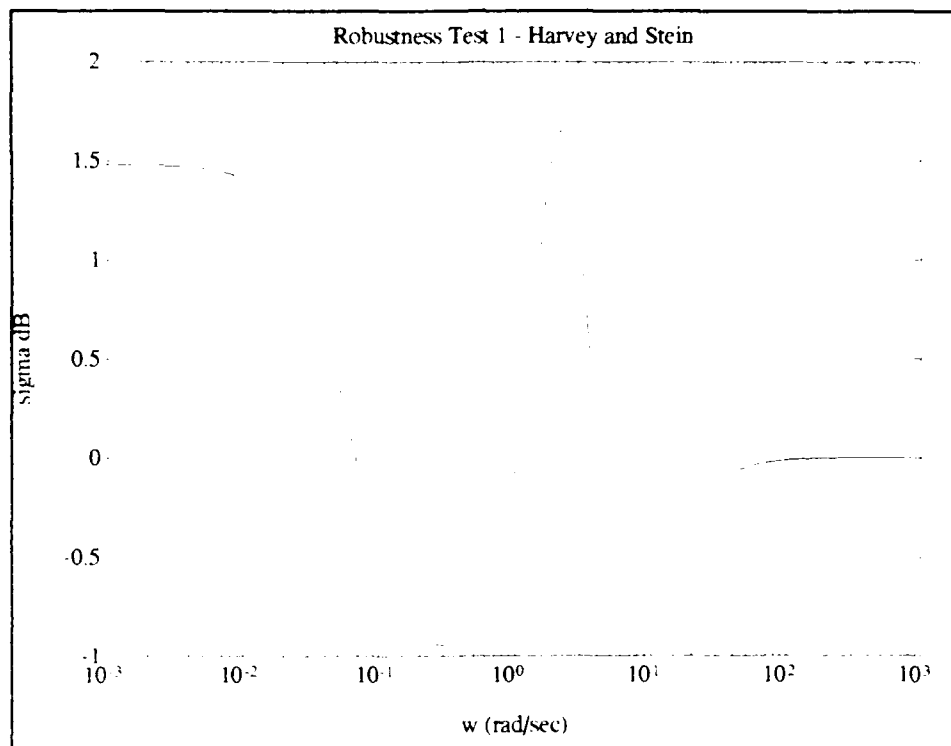


Figure 33 Minimum Singular Value Test 1 for Harvey and Stein Example

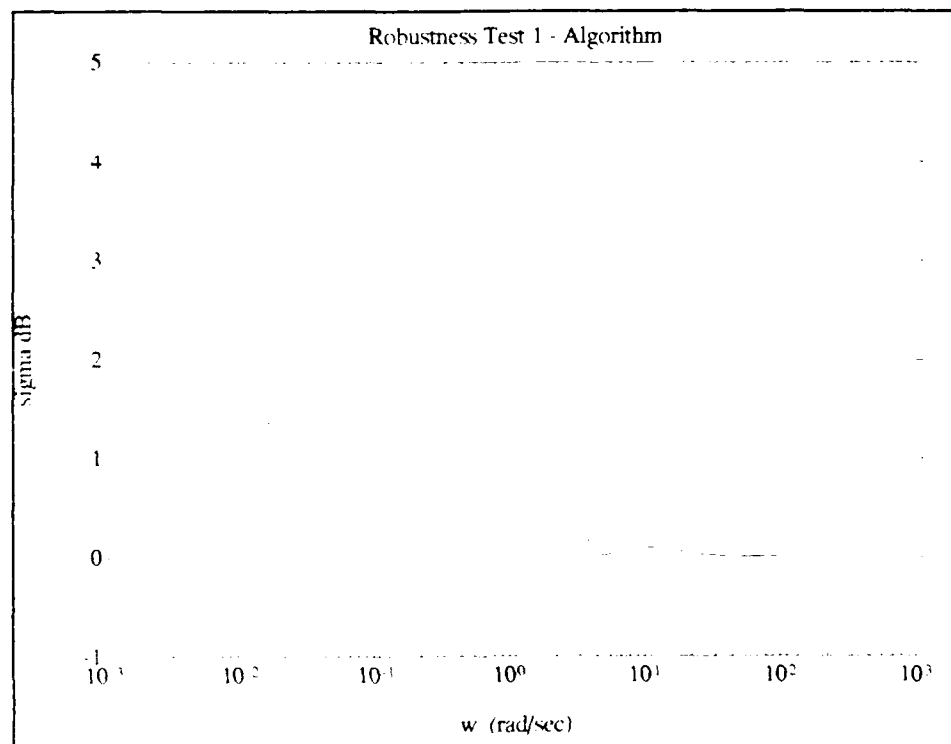


Figure 34 Minimum Singular Value Test 1 for Algorithm

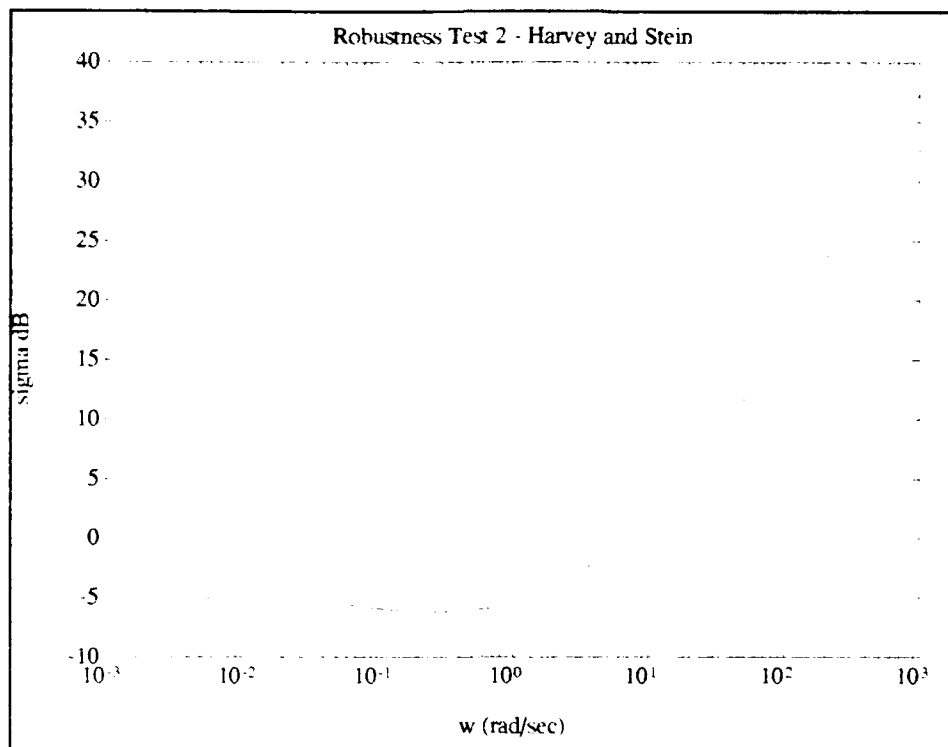


Figure 35 Minimum Singular Value Test 2 for Harvey and Stein

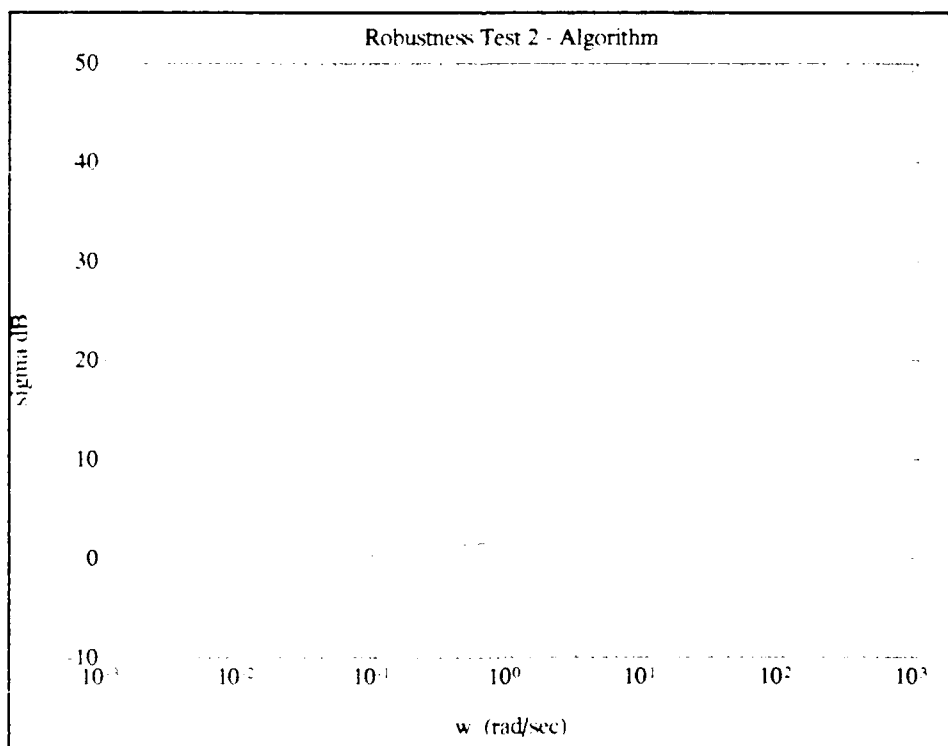


Figure 36 Minimum Singular Value Test 2 for Algorithm

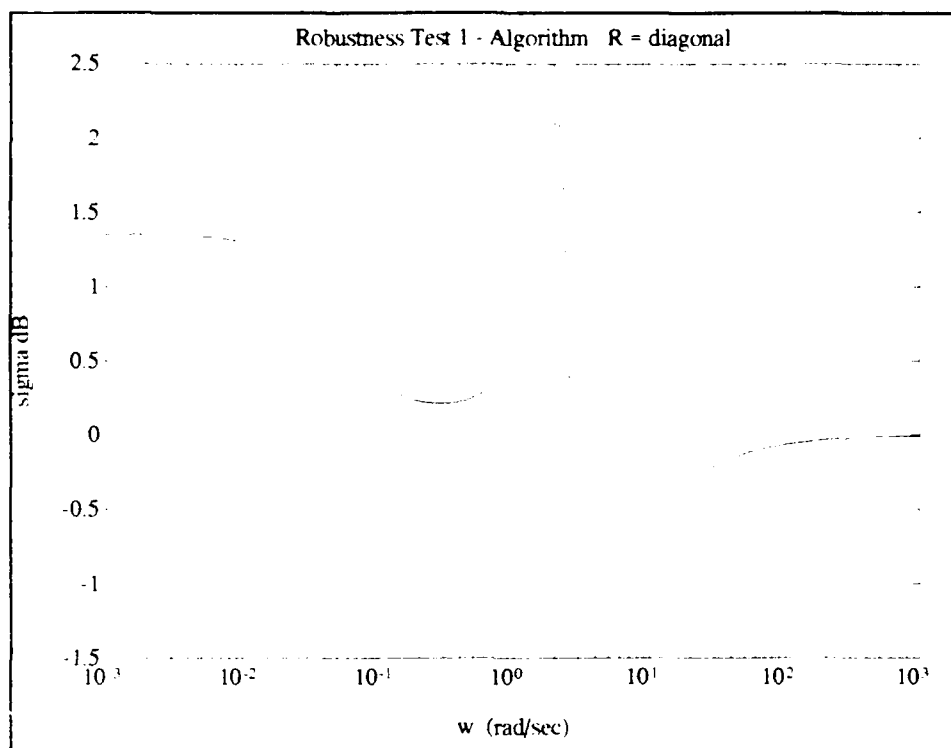


Figure 37 Singular Value Test for a Diagonal R

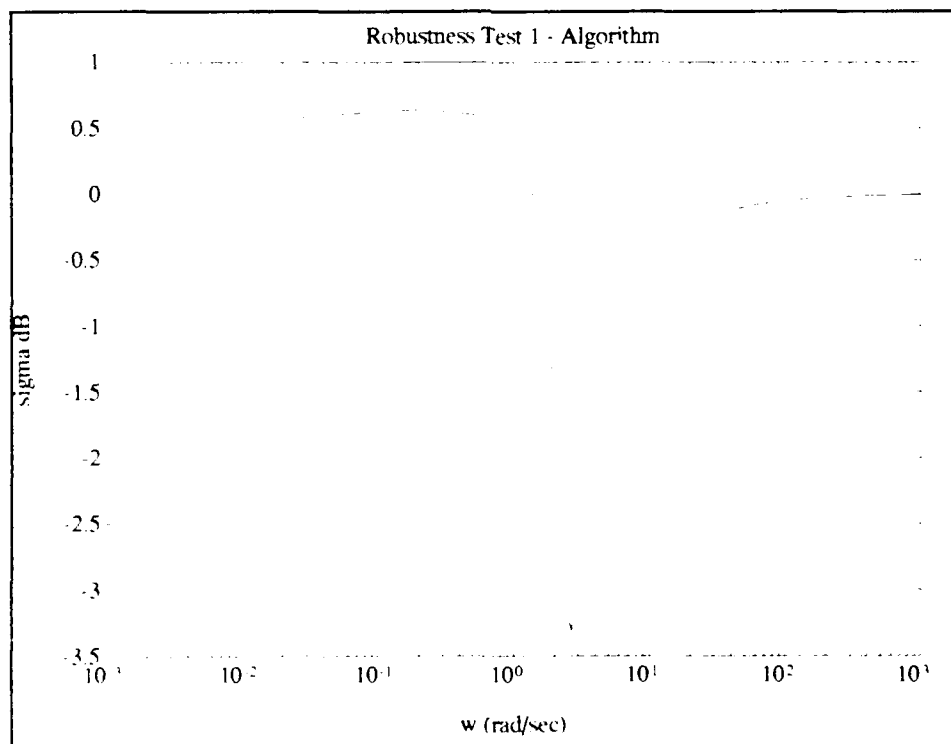


Figure 38 Singular Value Test for a Symmetric R

$$\mathbf{R} = \begin{bmatrix} 0.2424 & 0 \\ 0 & 0.8630 \end{bmatrix}$$

is given in figure 37. The IPM is reduced to $(-50.4^\circ, 50.4^\circ)$, well below the guaranteed minimum of $(-60^\circ, 60^\circ)$. However, there is a benefit for the reduction in stability margins - the achievable poles were placed exactly in the desired locations. Using a diagonal \mathbf{R} also gives one more parameter to design with. Let's go one step further. What if \mathbf{R} was allowed to be symmetric? This would give three variables in \mathbf{R} that could be changed, thus adding two extra degrees of freedom to the original problem. The algorithm was modified to let \mathbf{R} be symmetric and returned

$$\mathbf{R} = \begin{bmatrix} 0.0805 & 0.0581 \\ 0.0581 & 0.0563 \end{bmatrix}$$

The first robustness test in equation (98) is plotted in figure 38 for this symmetric \mathbf{R} matrix. All the poles were placed in the desired location, but the margins were reduced even further; the IPM = $(-40^\circ, 40^\circ)$ and the IGM = $(-4.5 \text{ dB}, 9.97 \text{ dB})$. If the designer had some freedom to relax the stability margins, a choice of a diagonal or symmetric \mathbf{R} matrix could be used to place all the poles in their desired locations. But some care must be taken if $\mathbf{R} \neq \rho \mathbf{I}$. If any one value in \mathbf{R} is much lower than the others, the corresponding parameter in the control vector would have unacceptably large input amplitudes [7:124]. The cross coupling also has constraints. If \mathbf{Z} represents unidirectional cross coupling, then for a diagonal \mathbf{R}

$$\overline{\sigma}^2(Z) < \frac{\lambda_{\min}(\mathbf{R})}{\lambda_{\max}(\mathbf{R})}$$

This is mentioned to highlight some of the potential problems that occur when $\mathbf{R} \neq \rho \mathbf{I}$. In this example all the elements in \mathbf{R} are of the same order, but each design case is different. The pole placement accuracy could outweigh the problems encountered with a diagonal or symmetric \mathbf{R} .

The Harvey and Stein technique placed the zeros in the desired pole locations by fixing \mathbf{Q} and varied \mathbf{R} to move the poles. The algorithm allowed both \mathbf{Q} and \mathbf{R} to vary, and achieved better results in both pole placement and robustness. Harvey and Stein's results shown in table V were calculated for their modified \mathbf{A} matrix. If their gains were applied to the actual aircraft \mathbf{A} matrix, their actuator pole locations would be further left and the stability margins would be worse. This example again demonstrates the pole placement accuracy using the method presented in this report. In the next example, an aircraft with open loop level II flying qualities is examined to see if the algorithm can improve flying qualities with a robust controller.

Example 5 - Aircraft Flying Qualities Improvement

This example demonstrates the algorithm's ability to improve the flying qualities of an aircraft. The aircraft in this example is a fictitious model similar to an A-4D, with characteristics primarily obtained from reference [21]. Appendix B lists the stability derivatives used in this example. Considering the longitudinal axis with only the elevator deflection as input, the state space representation is

$$\dot{\mathbf{x}} = \begin{bmatrix} -0.0129 & -3.7292 & 0 & -32.2 \\ -0.0002 & -0.8167 & 0.9984 & 0 \\ -0.0003 & -1.6903 & 0.0563 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 0 \\ 1.56 \\ 0 \end{bmatrix} u$$

where

$$\mathbf{x} = \begin{bmatrix} u \\ \alpha \\ q \\ \theta \end{bmatrix} \begin{array}{l} \text{velocity} \\ \text{angle of attack} \\ \text{pitch rate} \\ \text{pitch angle} \end{array} \quad (99)$$

and

$$u = \delta_e \quad (\text{elevator deflection}) \quad (100)$$

The open loop poles for this system lie at

$$-3.84 \pm 1.19j \quad (\text{short period})$$

$$-0.0026 \pm 0.03971j \quad (\text{phugoid})$$

which give

$$\zeta_{sp} = 0.307 \quad \omega_{n\,sp} = 1.25$$

$$\zeta_{ph} = 0.065 \quad \omega_{n\,ph} = 0.04$$

This flight profile (cruising flight) is classified as category B by the Mil-Spec users guide [22:14]. For category B, a level I flying qualities rating is defined when $0.30 < \zeta_{sp} < 2.0$ and $\zeta_{ph} > 0.04$. The short period frequency requirements are given in figure 39. The regions of flying qualities levels shown in figure 39 are for n/α , where n is the load factor [23:538-540]. Roskam [23] gives an approximation to n/α as

$$\frac{n}{\alpha} \approx \frac{-Z_{\alpha}}{g} \quad (102)$$

where $Z_{\alpha} = Z_{w,u}$. With the data in appendix B, n/α is found to be 16.11 g's/rad. Plotting ω_n vs 16.11 in figure 39 falls within the level II region (point A). The values for ζ_{sp} and ζ_{ph} fall within the level I range.

The goal is to make the system both level I and robust with full state feedback. Thus, the desired poles were chosen to give level I damping and place the frequency requirement in the center of the level I region. Point B in figure 39 corresponds to the desired short period frequency. The desired poles and associated ζ and ω_n are

$$-1.12 \pm 3.50j \quad \zeta_{sp} = 0.305 \quad \omega_{n\ sp} = 3.67$$

$$-0.0056 \pm -0.073j \quad \zeta_{ph} = 0.076 \quad \omega_{n\ ph} = 0.073$$

Using PLACE on MATLAB, the unique gains were calculated that gave the Bode plot for the desired poles in figure 40. The gain and phase margins from figure 45 are $GM = \infty$ and $PM = 38^\circ$. The GM is fine, but the PM is not quite high enough. The algorithm can not achieve the desired poles but it should get close. When this system was run on MATLAB with the algorithm m-files, the achievable poles were found to be

$$-2.096 \pm 2.389j \quad (\text{short period})$$

$$-0.215 \pm 0.043j \quad (\text{phugoid})$$

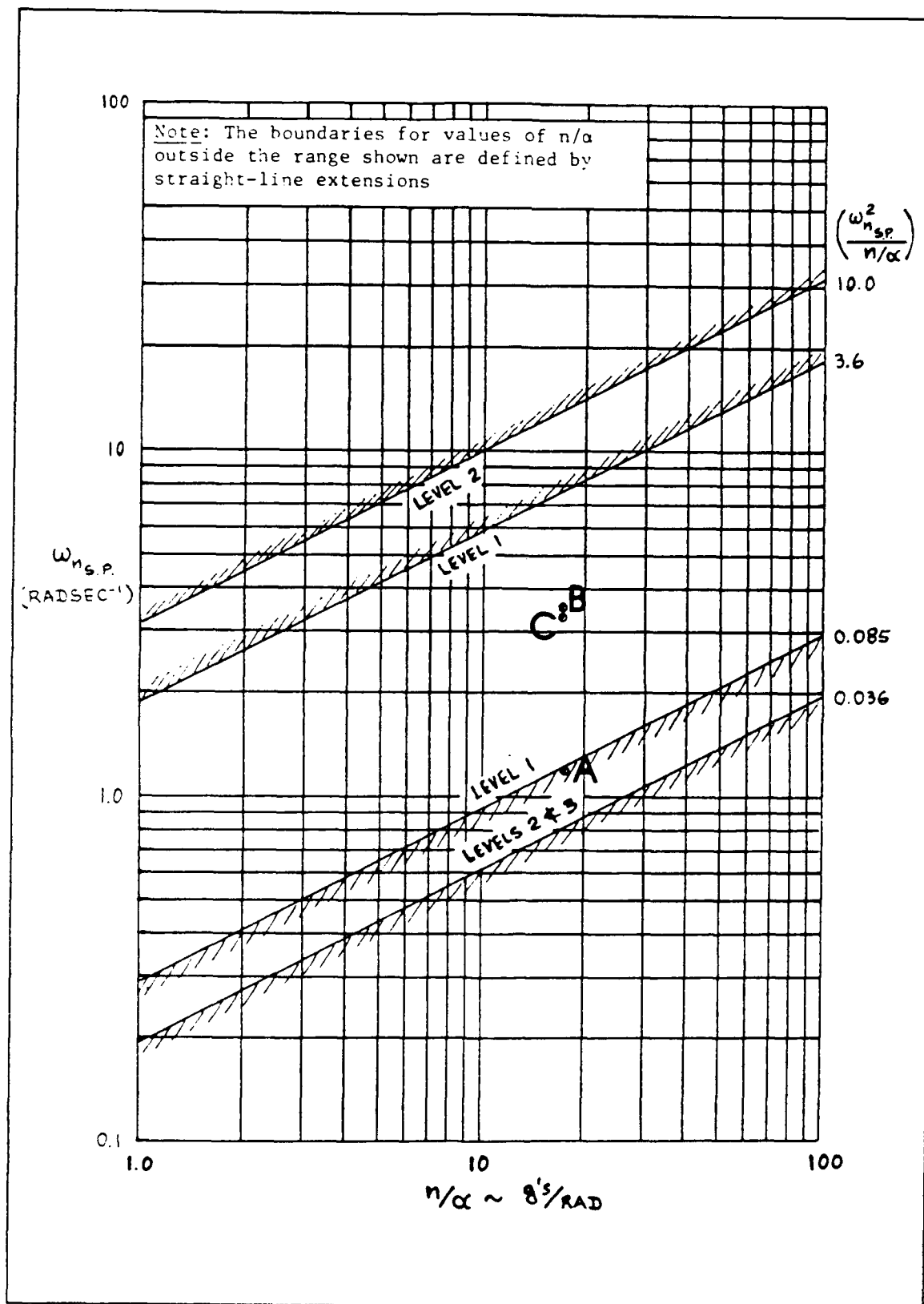


Figure 39 Short Period Frequency Requirements - Category B [23:541]

The natural frequency and damping are

$$\zeta_{sp} = 0.6595 \quad \omega_{n\ sp} = 3.179$$

$$\zeta_{ph} = 0.98 \quad \omega_{n\ ph} = 0.219$$

which are all within the level I region and close to the desired values. In figure 39, point C shows the short period frequency location for the achievable poles. The achievable system Bode plot shown in figure 41 gives the $GM=\infty$ and the $PM=68^\circ$. Thus, using the algorithm for pole placement gave a robust level I system.

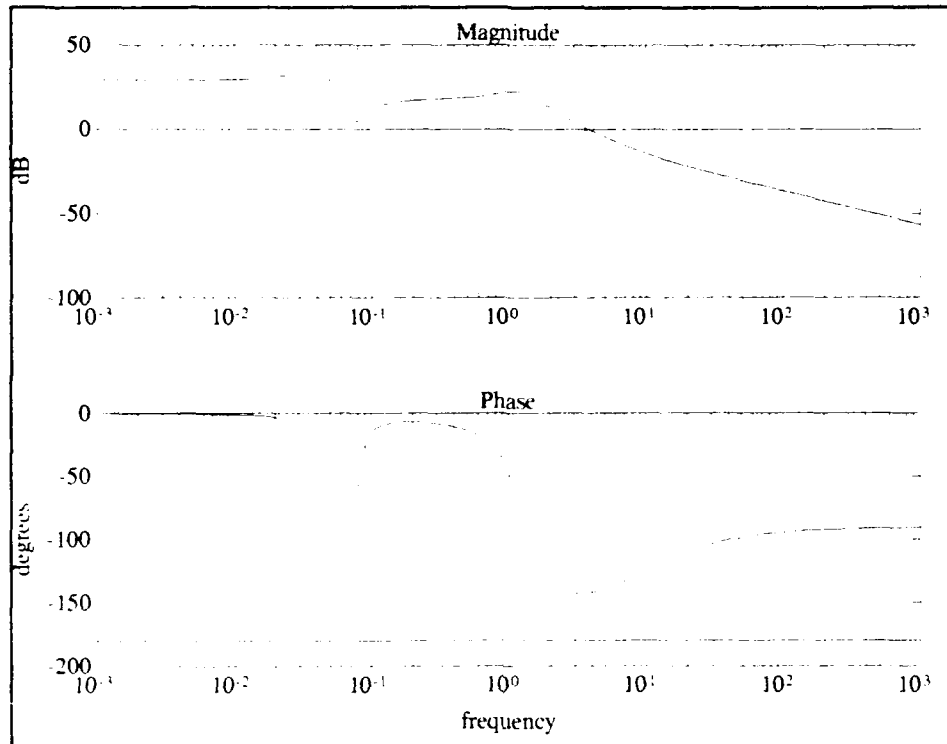


Figure 40 Bode Plot for Desired Poles for Example 5 Aircraft

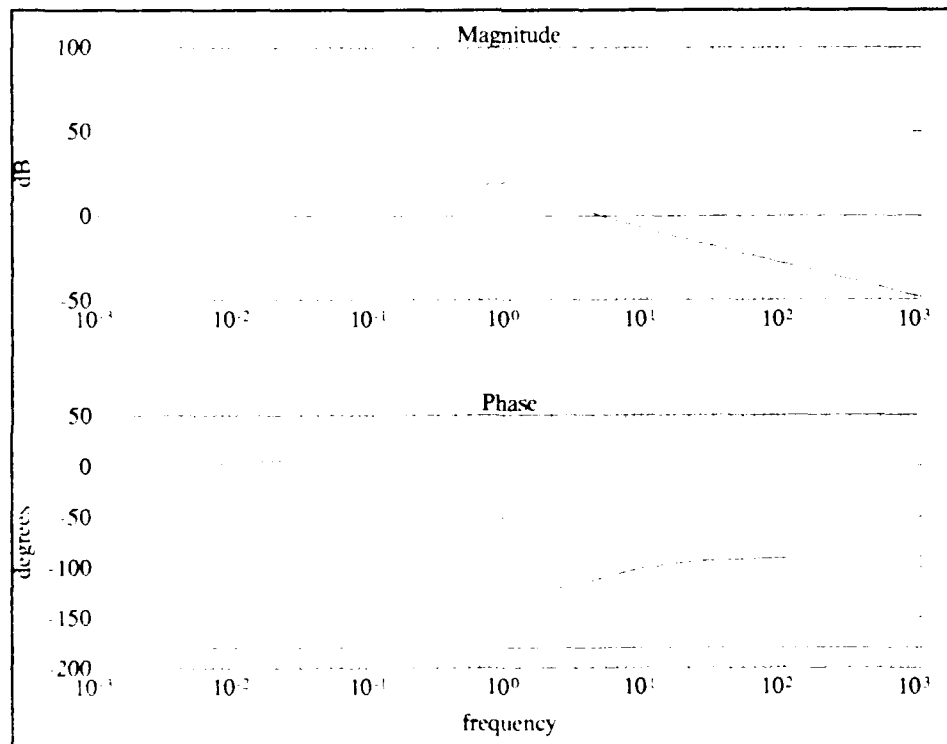


Figure 41 Bode Plot for Achievable Poles for Example 5 Aircraft

V. Conclusions and Recommendations

This report has examined the pole placement properties of linear quadratic regulators as determined by the weighting matrices of the performance index. A new eigenvalue difference cost-function to minimize the difference between sets of achievable poles and desired poles was introduced. This led to the formulation of an algorithm that employed a gradient search to optimize the eigenvalue difference performance-criterion. This algorithm gives a powerful method for selecting weighting matrices for pole placement designs.

Five examples were used to demonstrate the algorithm. The use of simple systems in the first two examples allowed closed form solutions which compared exactly to the solutions of the algorithm. The third order example showed the effects of weighting a desired pole. The heavier weighted pole was forced closer to the desired pole at the expense of the less weighted poles. A method presented by Harvey and Stein was compared to the algorithm given in this report, and the results show that the algorithm accurately provided robust eigenvalues for a MIMO system. In the last example, the algorithm was used to improve aircraft flying qualities. These examples showed that the algorithm can simultaneously place desired poles and achieve stability robustness.

This algorithm only employed an eigenvalue optimization performance criterion. The effects of optimizing both an eigenvalue and an eigenvector difference on achievable pole location should be investigated. The full effects of the LQR weighting matrices on pole locations is still not well understood. This algorithm could be modified to perturb selected values in the weighting matrices to see the effects on the closed loop poles. One

improvement that could be made is in the gradient search routine. The algorithm simply uses the Nelder-Mead method on MATLAB which calculates gradients numerically, but a more efficient method could be found. A possible way to increase the efficiency is to use the eigenvalue difference weighting parameter to automatically weight the poles with the largest eigenvalue difference. The m-files could undoubtedly be programmed more efficiently, and a Fortran source code would be useful to users who do not have access to MATLAB. This method is applicable to many flight control system design problems and could be used to improve the robustness properties of existing systems.

Appendix A. MATLAB m-files

Appendix A includes listings of the "m" files used on MATLAB to run the examples presented in this report. The following MATLAB functions were also used but are not included in this appendix:

FMINS - The gradient search minimization function in the MATLAB files.

PLACE - Used to find the gains required to place a pole at a desired location. This function is in the Control System Toolbox files.

MARGIN - This function finds the gain and phase margins for a given system. It is in the Control System Toolbox files.

LOGSPACE - A function in the MATLAB files used to generate a logarithmically spaced frequency vector.

BODE - Located in the Control System Toolbox file and used to generate Bode plots.

function [ea,k,Q,R]=LQRPP(A,B,ed,V)

```
% LQRPP      Pole placement using Linear Quadratic Regulator design.
%      [ea,k]=LQRPP(a,b,ed) will attempt to place poles in a
%      desired location (ed). If the desired location is outside
%      the allowable LQR region, LQRPP minimizes the distance
%      between the desired poles (ed) and the achievable poles (ea).
%      A desired pole can be weighted heavier to indicate its
%      relative importance. LQRPP will give a weighted pole
%      priority in the pole placement routine. The technique is to use
%      a gradient search to minimize the difference between the
%      desired and LQR achievable poles for the system
```

```

%           $\dot{x} = Ax + Bu$  with full state feedback  $u = -Kx$ 
%
% The required inputs are
%   A - the state space A matrix
%   B - the state space B matrix
%   ed - a vector containing the desired pole locations
%
% There is one optional input
%   V - the pole weighting matrix (diagonal)
%
% There are four output arguments (the last two are optional)
%
%   ea - a vector containing the achievable poles
%   K - the feedback gain matrix to get the achievable poles
%   Q - the LQR state weighting matrix
%   R - the LQR control weighting matrix
%
startpp
a_s=A;
b_s=B;
ed_s=ed;
[n,m]=size(A);
[l,p]=size(B);
% assign initial values to v, q, and r
if nargin > 3, v_s=V;
    else, v_s=eye(n); end
h=ppinit(n);
m=1; % initial value for rho
x=[h m];
plotinit(n,ed_s)
H=fmins('ppfunc',x,1e-2);
for i= 1:n
    plot(real(ea_s(i)),imag(ea_s(i)),'og')
end
title('Gradient Search For Achievable Poles')
xlabel('real')
ylabel('imag')
pause
k=k_s;
ea=ea_s;
Q=q_s;
R=r_s;
hold off

```

function J=ppfunc(x)

```
% used by LQRPP to find the value of the eigenvalue difference cost function
% This function calculates  $J$  and is called by the gradient search (FMINS) routine.
%

[s,t]=size(x);
[n,ma]=size(a_s);
[nb,mb]=size(b_s);
[nd,md]=size(ed_s);
xh=x(1:(s-1),1);
xm=x(s,1);
m=xm*eye(mb);
h=ppmakeh(n,xh);
q_s=h'*h;
r_s=m'*m;
k_s=lqr2(a_s,b_s,q_s,r_s);
ea_s=sort(eig(a_s-b_s*k_s));
%find the poles that are closest to each other and take their difference
eatmp=ea_s;
edtmp=sort(ed_s);
i=1;
J=0;
while i < n
    [nn,mm]=size(eatmp);
    z=abs(eatmp(1)*ones(nn,mm)-edtmp);
    [vt,is]=min(z);
    % Find if there are any weighted poles
    err=(ed_s-edtmp(is)*ones(nd,md));
    [vp,ip]=min(abs(err));
    J=J+vt^2*v_s(ip,ip);
    eatmp=eatmp(2:nn);
    if is==1, edtmp=edtmp(2:nn);
    else if is==nn, edtmp=edtmp(1:(nn-1));
    else, edtmp=[edtmp(1:is-1);edtmp(is+1:nn)];
    end;end
    i=i+1;
end
err=(ed_s-edtmp(1)*ones(nd,md));
[vp,ip]=min(abs(err));
J=J+(abs(eatmp-edtmp))^2*v_s(ip,ip);
for i=1:n
    plot(real(ea_s(i)),imag(ea_s(i)),'r')
end
```

```

%      startpp
%
%      Used by LQRPP to initialize the global variables used in the function

```

```

j=sqrt(-1);
global a_s b_s k_s q_s r_s ea_s ed_s v_s

```

function h=ppmakeh(n,x)

```

%
%      PPMAKEH is used by LQRPP to make the symmetric h matrix from a vector
%      n is the size of the matrix and x is a vector that contains the upper triangular
%      elements of the h matrix
s=1;
for i=1:n;
    for j=i:n;
        h(i,j)=x(s);
        h(j,i)=x(s);
        s=s+1;
    end
end
end

```

function plotinit(n,ed)

```

%      used with LQRPP to set up the plotting feature

axis('square')
e=sort(ed);
if abs(real(e(n))) > abs(imag(e(n))), axisize=ceil(abs(real(e(n))))+2;
    else axisize=ceil(abs(imag(e(n))))+2;
end
axis([-axisize axisize -axisize axisize])
plot([0 0],[-axisize axisize],'-b',[-axisize axisize],[0 0],'-b')
hold on
for i=1:n
    plot(real(e(i)),imag(e(i)),'xg')
end

```

function [f]=ppinit(n)

% used with LQRPP to initialize the weighting matrices
%

```
h=n;  
x=1;  
for i=1:n  
    f(x)=1;  
    for j= 1:(h-1)  
        x=x+1;  
        f(x)=0;  
    end %j loop  
    h=h-1;  
    x=x+1;  
end %i loop
```


Appendix B. Data for the A-4D Aircraft

The following data is for an A-4D flying in level flight at 15,000 ft at mach 0.6.

u_1 (ft/sec)	-	634.2
W (lb)	-	17,578
T_u (1/sec)	-	0.000225
X_u (1/sec)	-	-0.01288
X_w (1/sec)	-	-0.00588
$X_{\dot{w}}$ [(ft/sec ²)/rad]	-	0
Z_u (1/sec)	-	-0.1012
$Z_{\dot{w}}$ (-)	-	-0.001616
Z_w (1/sec)	-	-0.818
$Z_{\dot{w}}$ [(ft/sec ²)/rad]	-	0
M_u (1/sec-ft)	-	-0.000407
$M_{\dot{w}}$ (1/ft)	-	-0.000556
M_w (1/sec-ft)	-	-0.003
M_q (1/sec)	-	58.1
$M_{\dot{q}}$ (1/sec ²)	-	1.56

The dynamic equations of motion are given on the following page.

The equations of motion for the longitudinal axis are [23:172; 22:54]:

$$\dot{u} - u_1 X_w + g\theta = X_\delta \delta$$

$$(u_1 - u_1 Z_w) \dot{\alpha} - Z_u u - u_1 Z_w \alpha - (u_1 + Z_q) q = Z_\delta \delta$$

$$\dot{q} - M_u u + u_1 M_w \alpha + u_1 M_w \dot{\alpha} + M_q q = M_\delta \delta$$

$$\dot{\theta} = q$$

Bibliography

1. Blakelock, J. H. *Automatic Control of Aircraft and Missiles*. John Wiley & Sons, Inc., 1965.
2. McRuer, Duane T. and Thomas T. Myers. *Advanced Piloted Aircraft Flight Control System Design and Methodology Volume I: Knowledge Base*. NASA CR-181726, Langley Research Center, Hampton, Virginia, Contract NAS1-17987, October 1988 (N89-26013).
3. Wonham, W. M. "On Pole Assignment in Multi-Input Controllable Linear Systems," *IEEE Transactions on Automatic Control*, Volume AC-12: 660-665 (December 1965).
4. Moore, B. C. "On the Flexibility Offered by the State Feedback in Multivariable Systems Beyond Closed Loop Eigenvalue Assignment," *IEEE Transactions on Automatic Control*, Volume AC-21: 689-692 (October 1976).
5. Retallack, D. G. and others. "Pole-Shifting Techniques for Multivariable Feedback Systems," *PROC. IEE*, Volume 117: 1037-1038 (May 1970).
6. Simonyi, K. E. and N. K. Loh. "Robust Constrained Eigensystem Assignment," *American Control Conference Proceedings*: 960-965 (June 1989).
7. Anderson, Brian D. O. and John B. Moore. *Optimal Control, Linear Quadratic Methods*. New Jersey: Prentice Hall Inc., 1990.
8. Kawasaki, Naoya and Etsujiro Shimemura. "Determining Quadratic Weighting Matrices to Locate Poles in a Specified Region," *Automatica*, Volume 19: 557-560 (1983).
9. Wittenmark, Bjorn and others. "Constrained Pole-Placement using Transformation and LQ-design," *Automatica*, Volume 23: 767-769 (1987).
10. Graupe, D. "Derivation of Weighting Matrices Toward Satisfying Eigenvalue Requirements," *International Journal of Control*, Volume 16: 881-888 (1972).
11. Broussard, John R. "A Quadratic Weight Selection Algorithm," *IEEE Transactions on Automatic Control*, Volume AC-27: 945-947 (August 1982).
12. Solheim, O. A. "Design of Optimal Control Systems with Prescribed Eigenvalues," *International Journal of Control*, Volume 15: 143-160 (1972).

13. Wilson, Robert F. and James R. Cloutier. "Optimal Eigenstructure Achievement with Robustness Guarantees," *American Control Conference Proceedings*: 952-957 (1989).
14. Sobel, K. M. and E. Y. Shapiro. "Flight Control Examples of Robust Eigenstructure Assignment," *IEEE Conference on Decision and Control Proceedings*: 1290-1291 (1987).
15. Harvey, Charles A. and Gunter Stein. "Quadratic Weights for Asymptotic Regulator Properties," *IEEE Transactions on Automatic Control*, Volume AC-23: 378-387 (June 1978).
16. Kwakernaak, Huibert and Raphael Sivan. *Linear Optimal Control Systems*. New York: John Wiley & Sons, Inc., 1972.
17. Ridgely, Captain D. Brett and Siva S. Banda. *Introduction to Robust Multivariable Control*. AFWAL-TR-85-3102, Control Dynamics Branch, Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, February 1986 (AD-A165 891).
18. Safonov, Michael G. and Michael Athans. "Gain and Phase Margin for Multiloop LQG Regulators," *IEEE Transactions on Automatic Control*, Volume AC-22: 173-179 (April 1977).
19. Woods, D. J. *Report 85-5*. Department of Math. Sciences, Rice University, May 1985.
20. Franklin, Gene F. and others. *Feedback Control Dynamics Systems*. Massachusetts: Addison-Wesley Publishing Company, Inc., 1986.
21. McRuer, Duane T. and others. *Aircraft Dynamics and Automatic Control*. New Jersey: Princeton University Press, 1973.
22. Chalk, C. R. and others. *Background Information and User Guide For MIL-F-8785B(ASG), "Military Specification - Flying Qualities of Piloted Airplanes"*. AFFDL-TR-69-72, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, August 1969 (AD 860856).
23. Roskam, Jan. *Airplane Flight Dynamics and Automatic Flight Controls*. Kansas: Roskam Aviation and Engineering Corporation, 1979.

Vita

Jeffrey D. Robinson was born on 22 September 1960 in Alma Michigan, where he graduated from Alma High School in 1978. After enlisting in the U.S. Air Force in 1982, he worked in the simulator validation branch of the Electronic Warfare Center, Kelly A.F.B. In January 1984 he attended Auburn University in Auburn, Alabama under the Airman's Education and Commissioning Program, where he graduated with honors in 1986 with a degree of Bachelor of Aerospace Engineering. After receiving a regular commission in the USAF from OTS, Jeff served as a flight test engineer at Edwards AFB. While at Edwards he worked on the Anti-Satellite program (ASAT), the F-15E terrain following program, and the A-7F program. He then entered the School of Engineering, Air Force Institute of Technology, in June 1989.

Permanent Address: 4313 Birmingham Rd.
Alma, Mich. 48801

REPORT DOCUMENTATION PAGE			Form Approved OMB No 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE December 1990	3. REPORT TYPE AND DATES COVERED Master's Thesis		
4. TITLE AND SUBTITLE A Linear Quadratic Regulator Weight Selection Algorithm for Robust Pole Placement			5. FUNDING NUMBERS	
6. AUTHOR(S) 1Lt Jeffrey D. Robinson, USAF				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute Of Technology, WPAFB OH 45433-6583			8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/GAE/ENY/90D-23	
9. SPONSORING MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for public release; distribution unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) An algorithm is developed which selects the weighting matrices in the linear quadratic regulator design problem. The algorithm chooses the positive semidefinite state weighting matrix Q and the positive definite control weighting matrix R by attempting to place closed loop poles near a set of desired poles. The solution requires using a gradient search technique to minimize a weighted eigenvalue difference cost function. The weighting of the eigenvalue difference establishes the relative importance between the poles and thus forces selected poles closer to their desired location. The algorithm is run on the software package MATLAB, and the associated subroutines are discussed. Several examples are included to illustrate the use of the algorithm.				
14. SUBJECT TERMS Pole Placement; Eigenvalue; Flight Control Systems; Control Systems Optimization; Control Theory; Linear Quadratic Regulator			15. NUMBER OF PAGES 92	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	